RL-TR-94-213
Final Technical Report
December 1994

ROME LABORATORY

WIDE AREA NETWORKING R&D

NYSERNet, Inc.

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 6474 and 6797

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U.S. Government.

Rome Laboratory
Air Force Materiel Command
Griffiss Air Force Base, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-94-213 has been reviewed and is approved for publication.

APPROVED: Senatro J. Iuorno

SENATRO J. IUORNO
Project Engineer

FOR THE COMMANDER: John A Graniero

JOHN A. GRANIERO
Chief Scientist
Command, Control & Communications Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL (C3BC) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

WIDE AREA NETWORKING R&D

Richard Mandelbaum
James D. Luckett

Contractor: NYSERNet, Inc.
Contract Number: F30602-89-C-0016
Effective Date of Contract: 3 January 1989
Contract Expiration Date: 3 May 1993
Short Title of Work: Wide Area Networking R&D
Period of Work Covered: Jan 89 - May 93

Principal Investigator: Richard Mandelbaum
Phone: (718) 260-3050
RL Project Engineer: Senatro J. Iuorno
Phone: (315) 330-1873

Approved for public release; distribution unlimited.

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by Senatro J. Iuorno, RL (C3BC), 525 Brooks Rd, Griffiss AFB NY 13441-4505.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
December 1994

3. REPORT TYPE AND DATES COVERED
Final Jan 89 - May 93

4. TITLE AND SUBTITLE
WIDE AREA NETWORKING R&D

5. FUNDING NUMBERS
C - F30602-89-C-0016
PE - 62301E
PR - F474
TA - BO
WU - 01

6. AUTHOR(S)
Richard Mandelbaum and James D. Luckett

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NYSERNet, Inc.
Suite 103
200 Elwood Davis Road
Liverpool NY 13088-6147

8. PERFORMING ORGANIZATION REPORT NUMBER
NYSERNet-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington VA 22203-1714

Rome Laboratory (C3BC)
525 Brooks Rd
Griffiss AFB NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
RL-TR-94-213

11. SUPPLEMENTARY NOTES

Rome Laboratory Project Engineer: Senatro J. Iuorno/C3BC/(315) 330-1873

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This was a joint Rome Laboratory/ARPA advanced development effort that: (1) provided wide area high speed networking services for the New York State (NYS) Arpanet; (2) created the National Networking Testbed (NNET); and (3) developed Simple Network Management Protocol (SNMP) software tools. Under the first task, existing NYS Arpanet users were provided Transmission Control Protocol/Internet Protocol (TCP/IP) network service and support, access to the Defense Research Internet (DRI) with connectivity to the Internet, and upgraded service from 56 kbps to 1.544 mbps. The second task involved the installation and operation of the NNT, a nationwide, high-speed computer data network dedicated to ARPA's use to support domestic military and NASA sites, with access provided to the U.S. Dept. of Energy. The NNT provided a non-production network environment to field test a variety of network infrastructure devices and management/control software tools. The third task involved the development of network management software technology to operate and maintain reliable end-to-end NNT connectivity. In addition to SNMP software tools, some software development work involved applications related to the International Standards Organization Development Environment (ISODE) and X.500 Directory areas.

14. SUBJECT TERMS
High speed computer network, Internetworking, Wide area network research testbed, Transmission Control Protocol (TCP) (see reverse)

15. NUMBER OF PAGES
142

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

14. (Cont'd)

Internet Protocol (IP), Simple Network Management Protocol (SNMP)

Contents

Page

Appendixes

Figures

Tables

Preface

The work performed under this contract was based on a proposal submitted by NYSERNet, Inc. in response to RADC/DARPA Broad Area Announcement (BAA 88-1), issued June 20, 1988. The objective of this announcement was to solicit proposals for research in the area of advanced communications systems and distributed information sciences supporting Command, Control, and Communications (C3). NYSERNet's proposal called for the design, development, installation and operation of high speed transmission and switching component technology to upgrade a portion of the Arpanet and to provide the National Networking Testbed (NNT). The proposal also included a task to develop network management software technology to operate and maintain reliable end-to-end NNT connectivity.

Under contract to NYSERNet, Inc., Performance Systems International, Inc. (PSI, Inc.) managed development of the National Networking Testbed (NNT) and Simple Network Management Protocol (SNMP)/X.500 Directory research and development efforts. Another subcontractor, Williams Telecommunications Group (WTG) provided the components, facilities and telecommunications services required to build the NNT.

The principal investigator for the effort was Richard Mandelbaum of NYSERNet, Inc. The project director was James D. Luckett and Patricia Foster headed the management staff.

The project director at PSI was William L. Schrader. The project manager was Martin L. Schoffstall, who managed the telephone company line installations and decommissions. Mitch Levinn handled facility installations and operations. Equipment installations, point-of-presence (POP) construction, and site installations were managed by Barry Parlman. Engineering was accomplished by Messrs. Schoffstall, Levinn and Mark Fedor, Sr. The PSI research and development team was made up of Dr. Marshall Rose, Eric Jensen, Wengyik Yeong, and Christopher Kolb. They provided engineering, software development, and network management assistance.

The Rome Laboratory program manager was Senatro J. Iuorno. Lt. Col. Mark Pullen and Dr. Paul V. Mockapetris of DARPA provided technical guidance on the NNT architecture and SNMP/X.500 Directory aspects of the effort, respectively.

During the contract performance period both Rome Air Development Center (RADC) and the Defense Advanced Research Projects Agency (DARPA) experienced name changes. RADC became Rome Laboratory (RL) and DARPA was renamed Advanced Research Projects Agency (ARPA).

Abbreviations and Acronyms

ARC	Ames Research Center (NASA)
Arpanet	Advanced Research Project Agency Network
ARPA	Advanced Research Projects Agency
ASN	Abstract Syntax Notation
ATM	Asynchronous Transfer Mode
AT&T	American Telephone and Telegraph
BBN	Bolt, Beranek, and Newman, Inc.
BETA	Test of computer software under actual conditions
BOC	Bell Operating Company
bps	bits per second
CAN	Central Administration for the NADF
CCITT	International Telegraph and Telephone Consultative Committee
CONUS	Continental United States
CORNETT	A U.S. DOE network supported by the NNT
COSINE	Cooperation for OSI Networking in Europe
DAP	Directory Access Protocol
DARPA	Defense Advance Research Projects Agency
DARTNet	DARPA Advanced Research Testbed Network (one of the networks supported by the NNT)
DAS	Directory Assistance Service
DECCO	Defense Commercial Communications Office
DISA	Defense Information Systems Agency

Abbreviations and Acronyms (cont.)

DIT	Directory Information Tree
DLCI	Data Link Connection Identifier
DMD	Directory Management Domain
DNANS	Digital Equipment Corporation's (Digital Network Architecture) Name Service
DNS	Domain Name System
DOE	U.S. Department of Energy
DOS	Disk Operating System
DRI, DRI-1, DRI-2	Defense Research Internet, Defense Research Internet-1, Defense Research Internet-2
DSA	Directory System Agent
DSI	Defense Simulation Internet
DSO, DS1, DS3	Digital Systems 0, 1, and 3, which refer to the following signal levels of the time division multiplexing hierarchy: DS0 - 64 kbps (1 Voice Channel) DS1 - 1.544 Mbps (24 DSOs) DS3 - 44.736 Mbps (28 DS1s)
DUA	Directory User Agent
ESF	Extended Super Frame
ESNET	Energy Sciences Network (A U.S. DOE network supported by the NNT)
Ethernet	A local area network standard for the hardware and data link levels. There are two types: Digital/Intel/Xerox (DIX) and IEEE 802.3.
FDDI	Fiber Distributed Data Interface (100 Mbps)
FOX	Field Operational X.500

Abbreviations and Acronyms (cont.)

fr	An SNMP tool to monitor Frame Relay systems using an approach similar to SNMP Poll
FTP	File Transfer Protocol
Gb	gigabit
GFE	Government Furnished Equipment
IAB	Internet Activities Board
IDEAS	Documents which have since been replaced by "Internet Drafts"
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISI	Institute for Scientific Information
ISO	International Standards Organization
ISODE	International Standards Organization Development Environment
ITU-T	International Telecommunication Union (formerly CCITT) - Telecommunication Standardization Sector Recommendations
KAN	Knowledge and Naming
kbps	kilobits per second
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LDBP	Lightweight Directory Browsing Protocol
Los Nettos	A DARPA network supported by the NNT
Mbps	Megabits per second
MIB	Management Information Base

Abbreviations and Acronyms (cont.)

MILDEP	Military Department
MOA	Memorandum of Agreement
MS-DOS	Microsoft Disk Operating System
MTA	Message Transfer Agent
NADF	North American Directory Forum
NASA	National Aeronautics and Space Administration
NASNET	Numerical Aerodynamic Simulation Network (NASA)
NIC	Network Information Center. Located at SRI International, it is the primary repository for RFCs and Internet Drafts, as well as providing other services.
NISC	Network Information and Support Center
NIST	National Institute of Standards and Technology
NNT	National Networking Testbed
NOC	Network Operations Center
NSF	National Science Foundation
NSI	NASA Science Internet
NTN	National Telecommunications Network
NYS	New York State
NYSERNet	New York State Education and Research Network
O&M	Operations and Maintenance
OS3	transmission at 155 million bps
OSI	Open Systems Interconnection (a set of protocols designed to be an international standard method for connecting unlike computers and networks)

Abbreviations and Acronyms (cont.)

OSI-DS	Open Systems Interconnection-Directory Service
PC	Personal Computer
PDU	Protocol Data Unit
POP	Point-of-Presence
PSI	Performance Systems International, Inc.
QoS	Quality of Service
QUIPU	A complete implementation of the Directory from the University College of London
RADC	Rome Air Development Center
R&D	Research and Development
RFC	Request For Comments
RL	Rome Laboratory
RR	Resource Record
SIGCOMM	Special Interest Group on Data Communications
SMDS	Switched Multimegabit Data Service
SNMP	Simple Network Management Protocol
SONET	Synchronous Optical Network
T1, T3	Refers to the type of digital carrier facility used to transmit the following formatted digital signals: T1 - 1.544 Mbps (DS1) T3 - 44.736 Mbps (DS3)
TCP/IP	Transmission Control Protocol/Internet Protocol
TWBnet	Terrestrial WideBand Network (a DARPA network supported by the NNT)
UFN	User Friendly Naming

Abbreviations and Acronyms (cont.)

WAN Wide Area Network

WPP White Pages Project

WHOIS An Internet program which allows users to query a database of people and other Internet entities, such as domains, networks, and hosts, kept at the NIC. The information for people shows a person's company name, address, phone number and email address.

WTG Williams Telecommunications Group

X.400 Refers to the ITU-T (formerly CCITT) X.400 Series of Recommendations which define the standard architecture, services and protocols for Message Handling Systems.

X.500 Refers to the ITU-T (formerly CCITT) X.500 Series of Recommendations which define the system models, service elements and protocol aspects of the Directory Access Service, the Directory System Service and the Authentication Framework required for a complete Directory Service.

Summary

This contract resulted in the design, development and implementation of high speed, wide area networking services for the New York State (NYS) Arpanet and the National Networking Testbed (NNT). It also resulted in the development of prototype software applications designed to control and manage these networks. The effort covered 3 major tasks.

Under the first task, NYS Arpanet traffic was re-routed over NYSERNet, a high speed, wide area network within NYS with links to most national networks. This new high speed access provided existing NYS Arpanet users Transmission Control Protocol (TCP)/Internet Protocol (IP) service and support, access to the Defense Research Internet (DRI) with connectivity to the Internet, and up-graded service from 56 kbps to 1.544 mbps.

The second task involved installation and operation of the NNT, a nation-wide, high speed (T1 and T3 capabilities) computer data network facility dedicated to DARPA's use to support domestic military and NASA sites, with access provided to the Department of Energy. The NNT afforded increased bandwidth and flexibility to research networks. It provided a non-production research environment to field test both hardware, such as routers, switches, and related experimental and prototype communications equipment, and network management, control and tracking software tools. A variety of network infrastructure devices and software tools were developed and tested using the NNT, both at the points-of-presence (POPs) and at the connecting sites. The NNT consisted of 6 different backbones across the United States and at one time supplied service to nearly 100 sites connected to the following networks: Arpanet, Los Nettos, CORNETT, ESNet, Terrestrial Wideband Network (TWBnet), and the DARPA Advanced Research Testbed Network (DARTNet). The NNT experienced extensive use over a period of 4 1/2 years. The U.S. Dept. of Energy employed the network for two of its systems, ESNet and CORNETT.

The third task involved the development of Simple Network Management Protocol (SNMP) software tools and other software tools related to network infrastructure, such as International Standards Organization Development Environment (ISODE), related application software areas such as X.500 Directory, and prototype demonstrations of those software application tools. Many of the developed network management software and computer network applications software tools were installed and tested over the networks operated on the NNT by various researchers at the connected sites.

It was concluded that the POP-based switching and routing approach is capable of sustaining a nationwide high speed computer network.

Specialists should be employed to use the standard equipment of the telephone companies to advance research and development. Additionally, federal agencies should be allowed the flexibility in a network to direct their own activities.

Section 1
Introduction

This final technical report describes the services provided by NYSERNet, Inc. and its subcontractors, under the Wide Area Networking R&D contract. The subcontractors were Performance Systems International, Inc. (PSI, Inc.) and Williams Telecommunications Group (WTG). The original term of the contract was 24 months, from 3 January 1989 to 3 February 1991. The contract term was extended three times, twice for twelve months and the third time for two months. The first extension allowed the Government to take advantage of the capabilities of the installed networks for a longer period. The second extension allowed continued refinement of the networking technology developed under the contract in preparation for its transition to the Defense Information Systems Agency (DISA). The final 2-month extension was necessary to accomplish the network technology transition and required contract closeout actions (finalize circuit reconciliation costs, POP hardware rental costs, and plans for disposing residual equipment). Contract performance was completed on 3 May 1993.

1.1 Objectives

The objectives of this effort were to design, develop, install and operate high speed transmission and switching component technology to upgrade the New York State (NYS) Arpanet and to provide the National Networking Testbed (NNT); and to develop network management software technology to operate and maintain reliable end-to-end NNT connectivity.

1.2 Background

High speed data networks are becoming an essential tool in scientific research, engineering, administration, finance, and other areas that use extensive amounts of data. The efficient movement of data results in the following benefits:

- Insights gained by collaboration from participants located in diverse areas;

- Optimization of expenditures by providing timely access to information.

The growth in computing power is outstripping the abilities of networks in use today to transmit data.

Six years ago the research community doubted it would outgrow the 56,000 bits per second (bps) networks then in use. Today it worries that the 45 million bps networks now being installed will be sufficient for only a few years. If access to networks is not to be restricted to a few researchers, efforts are required today to reach 155 million bps (OC3) and higher speeds.

The Internet research community is comprised of many people and organizations in the U.S. government, universities, and industry. This community has an interest in the stable operation of the existing global research network. It also promotes the extension of this network to new regions and the development of new capabilities.

The Internet research community also interacts to identify and solve networking problems. Its diverse centers of strength can create competing and incompatible solutions to problems. Testing various environments and protocols on the existing networks can disrupt traffic. The networking community needs access to field test environments to test prototypes.

1.2.1 New York State Arpanet

In its services to DARPA and Rome Laboratory, PSI and NYSERNet, Inc. adapted NYSERNet - the network - for use as a new mechanism to access the Internet for sites in New York State and the Canadian Defense Research Network. This network was to connect to the Internet with the capability of transmission at 1.544 Mbps (T1) on shared bandwidth. The objective was to deliver access to the Defense Research Internet (DRI) from Arpanet sites at Cornell University, Rockefeller University, University of Rochester, Columbia University, and the Rome Laboratory and to provide this access through NYSERNet. Responsibility for delivering defense research-related network traffic would transfer, therefore, from Arpanet to NYSERNet. The change from Arpanet to NYSERNet was implemented to improve economy, performance, and service. Additional objectives were to provide TCP/IP (Transmission Control Protocol/Internet Protocol) network service. NYSERNet, Inc. and PSI provided support for the transition from the Arpanet to NYSERNet.

1.2.2 Development of NNT

The National Networking Testbed was created to minimize the delay between the creation of competing approaches and the operational use of workable solutions. The National Networking Testbed gives researchers a facility to use in place of the existing networks, a prototype that can be used without disrupting current users' traffic. NYSERNet developed the National Networking Testbed for the following specific purposes:

- Support, facilitate, and encourage research in the area of high speed networking technologies;

- Support wide area data communications networks.

The following government agencies participated in the National Networking Testbed: Rome Laboratory, Defense Advanced Research Projects Agency (DARPA), and the U.S. Department of Energy (DOE).

During the 1980s, government agencies invested millions of dollars each year to explore and prepare for data transmission speeds in the gigabit (Gb) range over fiber optic lines. Nevertheless, in 1988, most networks operated at only 56 kbps; a few operated at 1.544 megabits per second (Mbps), also known as T1. The National Networking Testbed was designed initially with six T1 circuits in the backbone. T1 tail circuits were installed from the backbone to numerous institutions. This immediately provided a network operating at speeds an order of magnitude faster than that available in 1988 and could easily have been upgraded to 45 Mbps.

The National Networking Testbed was designed for the following objectives:

- Increase the bandwidth available for industrial, academic, and government research networks;

- Increase the flexibility available to the networks;

- Achieve these objectives within the same general budget.

NYSERNet, Inc. was the lead organization in the team of not-for-profit and industrial participants in developing the National Networking Testbed. NYSERNet chose the National Telecommunications Network (NTN) to build the National Networking Testbed. NTN was a partnership of five inter-lata carriers that provides telecommunications services nationwide. In 1989, NTN turned over its contracts to Williams Telecommunications Group (WTG). WTG is an experienced telecommunications service provider that uses fiber optic facilities. WTG has operated a national network with reliable digitized fiber optic facilities. Its qualifications are summarized below:

- Its ability and willingness to collaborate within a complex research environment, allow access to their facilities (POPs, fiber plant, repeaters) and to monitor and alter equipment, software, and other parameters to permit full use of the NNT infrastructure.

- Its expertise, outstanding existing facilities, sound organizational structure and management permitted NYSERNet to deliver the bandwidth and provided almost unlimited growth potential for the future. Multiplexing options, dynamic reconfiguration multiplexing, and dark fiber availability were top among the technical advantages. The availability of 100% fiber, Extended Super Frame (ESF) with digital switching was important as it allowed NYSERNet to explore gigabit speeds. Its damage protection, stand-by services, and performance specifications were extremely high quality.

- It demonstrated the greatest flexibility in pricing options on all components of the project including lines, surcharges on local loops, multiplexing, installation, and operational support.

- It promised delivery on NYSERNet's schedule. All DS1s were to be cut over within 45 days. Local loops were not directly under their control, but it was prepared to work to insure a rapid cutover of these facilities.

1.2.3 Simple Network Management Protocol

The Internet Activities Board (IAB) designated the SNMP as the primary management tool for TCP/IP networks in 1992. The SNMP includes a number of supporting data structures and information data bases. Notable among them is its Management Information Base (MIB-II) and more focused MIBs including T1, T3, Ethernet, and FDDI.

1.3 Scope

The scope of this effort covered services to design, develop, install and operate high speed wide area networking services for the New York State (NYS) Arpanet and the NNT, and to develop SNMP software tools to operate and maintain reliable end-to-end NNT connectivity.

1.4 Task Schedule and Milestones

Figure 1 depicts the task schedule and milestones of the overall effort.

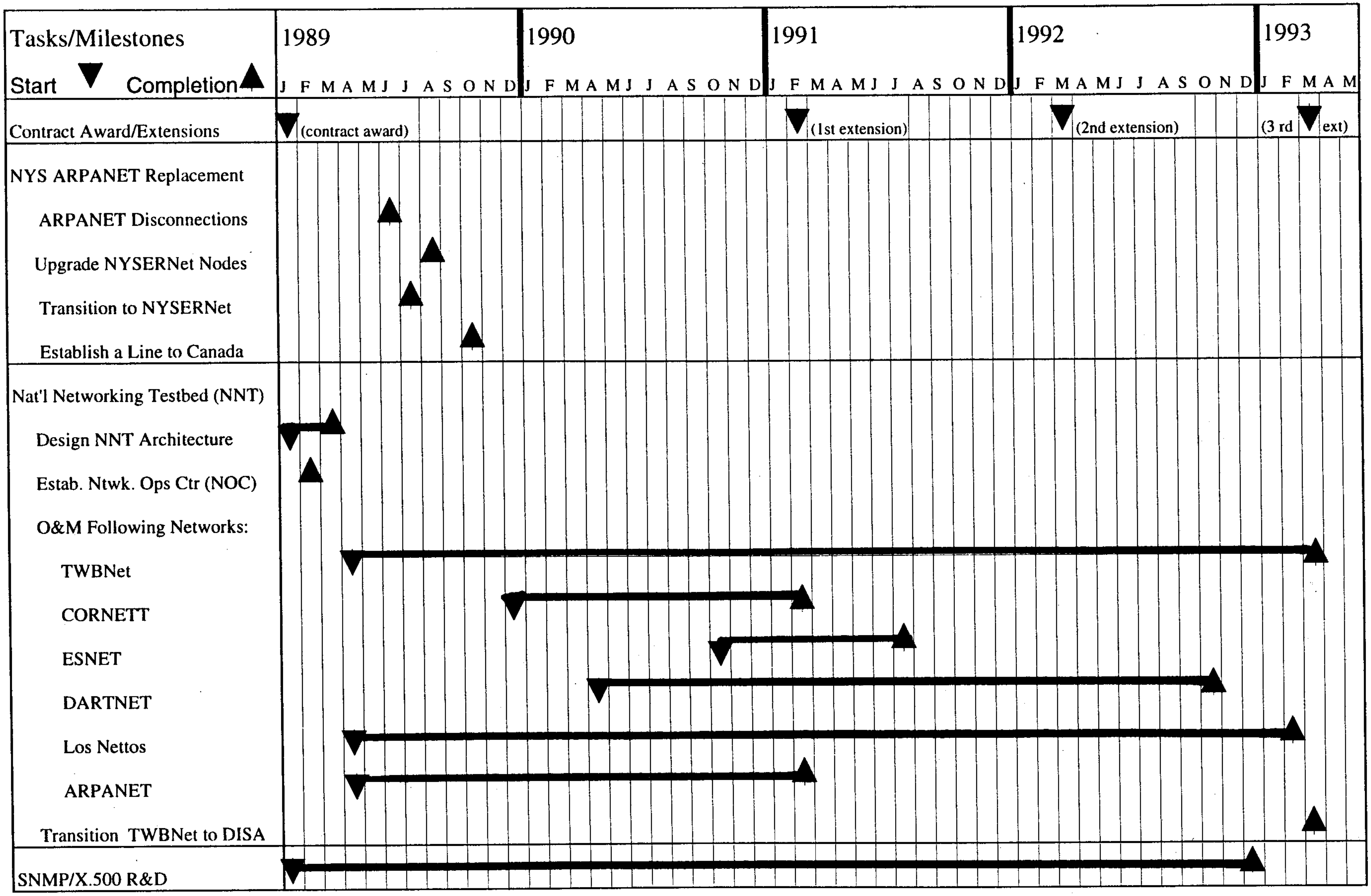


Figure 1. Task Schedule and Milestones

1.5 Report Content

This report is organized as follows:

- Section 1	- Introduction
- Section 2	- Network Development
- Section 3	- SNMP/X.500 Directory Software Development
- Section 4	- Conclusions and Recommendations
- Appendix A	- NNT Topology Supporting Six Networks (TWBnet, CORNETT, ESNET, DARTNet, Los Nettos, and Arpanet)
- Appendix B	- Current Participants in the White Pages Project
- Appendix C	- Application Monitoring MIB
- Appendix D	- The String Representation of Standard Attribute Syntaxes
- Appendix E	- Lightweight Directory Access Protocol
- Appendix F	- Comments on NADF Agreements For Name and Knowledge Sharing

Section 2
Network Development

This section addresses the approach used to design and deliver the operating Arpanet project and the National Networking Testbed (NNT).

2.1 New York State Arpanet

At the beginning of work under Contract F30602-89-C-0016, Cornell University, Columbia University, University of Rochester, and Rockefeller University were connected twice to the Internet - once through NYSERNet and once through Arpanet. The result was duplicate expense in facilities and management. Performance was to improve by routing traffic to the Internet over NYSERNet rather than Arpanet.

Each university had an Arpanet connection in the computer science department. The NYSERNet connection was in the campus computer center. The services provided would improve with the switch to NYSERNet.

The project required two equipment upgrades at the NYSERNet node at each university. The first upgrade provided Arpanet users with a second and completely independent local area network (LAN) interface in the node. Their traffic enters the Internet independently of the campus traffic on the other LAN interface. The second part of the upgrade increased the router's speed and enabled it to handle the added traffic.

The project required about 4 months to implement the transition in services of the Network Operations Center and the Network Information and Support Center. The transition services were also available to Rome Laboratory. These services educated users and systems' staff to see that routing and other network parameters were properly established.

2.1.1 DRI/Internet Access Approach

The following steps were taken in the approach to connect Cornell University, Rockefeller University, University of Rochester, and Columbia University to the Defense Research Internet (DRI) and to the Internet through the NYSERNet network.

a. Assessed the LAN and related LAN and host environment that would be used to connect to NYSERNet at the four campuses. Jointly determined a schedule of re-connection to NYSERNet.

b. Together with DARPA and Rome Laboratory, announced the upgrade of service from 56 kbps to 1.544 Mbps (T1) through NYSERNet.

c. Determined and implemented any required LAN alterations, host reconfigurations, network and address changes, and related technology modifications to connect the current hosts on the Arpanet to NYSERNet. To effect the transition, NYSERNet provided technical assistance and nominal materials and software.

d. Upgraded the NYSERNet nodes on the four campuses from a single Ethernet connection and a 68010 Proteon P4200 router to a twin Ethernet and a 68020 P4200. The planned NYSERNet 68020 P4200 node at Rome Laboratory was integrated into the Arpanet.

e. Developed and conducted on-site seminars and documentation to the local research community; announced the transition from Arpanet at 56 kbps to NYSERNet at 1.544 Mbps (T1). These quarterly seminars addressed IP and ISO; they also solicited reports on usage and problems.

f. Provided a Network Operations Center (NOC) to monitor network status and performance, identify troubles, repair the network, and provide a Root Domain Server.

g. Provided a Network Information and Support Center (NISC) as a technical point of contact for institutions and general user support.

h. With the involvement of Canadian personnel, assessed the Canadian defense LAN and wide area network (WAN) environments related to networking between Canada, Arpanet, and NYSERNet.

i. In cooperation with Canadian staff, took the following steps:

(1) Determined an implementation plan and provided installation of a 56 kbps line from the Buffalo NYSERNet node to Canada to furnish access to the Defense Research Internet.

(2) Provided a network infrastructure which could easily be upgraded to T1.

(3) Acquired and integrated a CISCO router and related line termination equipment at the existing node at SUNY Buffalo.

(4) Provided redundant routes from the Buffalo node to the U.S. Internet from the incoming 56 kbps Canadian line.

(5) Reconfigured routing appropriately and provided a spare router for emergency use on the U.S. side.

j. With DARPA and its contractors, assessed and coordinated related international WAN traffic for the New York State Arpanet and Canadian link to other U.S. and European networks. Provided effective routing, monitoring, and administrative coordination as required.

2.2 National Networking Testbed

At the outset, it was the intent to design the National Networking Testbed with a sound basis that was sufficiently flexible such that it could later be adopted as a fully functioning system. The topology selected placed the routing equipment at the end site; the multiplexers were placed in the POPs. Because the agencies using the testbed chose the routers, they were not included in the cost for this effort. It was necessary to work closely with the appropriate parties to assist with installation, management, and troubleshooting to provide for dependable and consistent operation of the National Networking Testbed.

A number of design alternatives were available for use in the National Networking Testbed. Several alternatives are listed below:

Extended NNT routes - National Telecommunications Network offered special low pricing on its T1 lines feeding the National Networking Testbed for each line designated as a future DS3 route; it also agreed to upgrade the lines to DS3 within 36 months.

Local Loop Bypass - A bypass was feasible in each city where it was available. This was an alternative to traditional services from the Bell Operating Company (BOC). The bypass options were not implemented.

Additional DS3 Cities - Many additional cities could have been connected to the DS3 backbone because they were on the path of the fiber. Little additional cost would have been incurred. Philadelphia and Baltimore are examples of such cities.

DS3/DS1 Multiplexing - Multiplexing took place in the POP. An alternative was multiplexing at the end site, but that option would have required additional DS3 loops, a large expense. Maintenance would also have been rendered more complex. The arrangement would also have reduced flexibility at the endsite. Therefore, the use of multiplexers from the inter-lata carrier accommodates more flexibility.

DS1/DS0 Multiplexing - Multiplexing from DS1 to DS0 was available, but it was seldom used. Most networks upgraded from 56 kbps to 1.544 Mbps (T1) once attached to the National Networking Testbed.

Point of Demarcation - The National Networking Testbed's demarcation point could have been at the POP, the end site, before or after the multiplexor, and before or after the router. It was, thus, truly flexible.

2.2.1 NNT Development Approach

The following steps were taken in the approach to design, develop, and implement the National Networking Testbed (NNT) architecture and to install, operate, test, and integrate high speed transmission and switching component technology:

a. Designed the detailed National Networking Testbed architecture. It consisted of six DS1 lines with appropriate DS3/DS1 multiplexing, switching tail circuits, and monitoring equipment.

b. Installed the DS1 configuration to nearly one hundred institutions that provided DS1 National Networking Testbed service to six separate national backbones.

c. Established and maintained the following:

(1) Documentation of the system

(2) Configuration management system with detailed specifications on the entire National Networking Testbed architecture

(3) System log books

(4) Hardware and software design handbooks

(5) Software listings

(6) Operator's manuals

(7) Logic diagrams

(8) Maintenance procedures

d. With WTG, determined and developed performance specifications and limitations, monitoring systems, trouble detection, and trouble resolution protocols.

e. Provided access to the WTG POPs on the National Networking Testbed for installation of the data communications and related equipment of the National Networking Testbed.

f. Designed, procured and equipped, integrated and tested the National Networking Testbed Network Operations Center at the NYSERNet Network Information and Support Center in Troy, New York. This Network Operations Center was integrated with the WTG National Network Monitoring Center in Tulsa, Oklahoma and the Network Operating Centers for the various components of the National Networking Testbed.

g. Provided 24-hr network surveillance and troubleshooting including services related to line outages, transient line and multiplexor problems, and, where possible, traffic statistics. The National Networking Testbed's Network Operations Center was available to provide network engineering recovery assistance to the agency Internet managers using the National Networking Testbed. NYSERNet provided access to test results, statistical and monitoring data to agency networks as needed. It also provided appropriate documentation.

h. Periodically inspected and tested portions of the National Networking Testbed using established procedures to see if the equipment was operating within the specified limits.

i. Isolated faults and took corrective actions when error conditions were found.

j. In the case of the Department of Defense (DoD) sponsored networks, directed and coordinated firms contracted to NYSERNet, Inc. for National Networking Testbed installation and operations with operational networks through their sponsoring agencies and those agencies' contractors. The operating networks, sponsoring agencies and support contractors included the following:

- DARPA's Arpanet
  Bolt, Beranek, and Newman, Inc. (BBN)
- Terrestrial WideBand Network (DARPA)
  BBN, PA Bell, NJ Bell, SW Bell, Pac Bell
- DARTNet
  ISI, BBN, NASA-Ames

These activities included specific network operations and extensions of network circuits to demarcation points. The Department of Energy handled these activities for their ESNET and CORNETT networks.

k. Designed and offered training seminars on National Networking Testbed configuration, operation, and management to staff at associated institutions, agencies, and networks.

l. Developed and implemented procedures to use Internet traffic on the National Networking Testbed. This was done to test processes and components of new data communications technology. The tests were conducted with minimal disruption.

m. Designed and implemented a flexible National Networking Testbed hardware environment. This environment was capable of field testing all types of data communications and Internet equipment. It included routers, switches, multiplexers, and fiber optic electronics.

n. Designed and implemented a flexible National Networking Testbed software environment. This environment was used to test and develop network management, control, tracking, and other statistical tools.

2.2.2 NNT Topology

The National Networking Testbed was designed initially with six T1 circuits in the network's backbone. The high speed backbone used T1 portions of a DS3 fiber optic system from WTG. Their purpose was to provide the capability for routine use and testing. Full-scale tests of the DS3 packet switches and some dark fiber experiments were possible.

The six T1 circuits also provided maximum flexibility in the backbone and permitted different sites to be connected to different circuits in the backbone. T1 tail circuits were installed from the backbone to numerous institutions. The result was an increase by an order of magnitude over the network's 1988 operating speed. The speed could easily have been upgraded to 45 Mbps.

In each of the POPs, DARPA installed routers and security devices from BBN. The capability to install, maintain, and change switches and routers with the POPs gave DARPA necessary flexibility.

The design backbone topology of the NNT is illustrated in Figure 2. The topology at each site is shown in Figure 3. The initial topology followed the fiber optic DS3 route between Boston, New York, Washington, Pittsburgh, Chicago, Denver, Los Angeles, San Diego, and from Los Angeles to San Francisco. At each of these cities (the "DS3 cities"), a DS3 to DS1 multiplexor was installed at each POP. This arrangement provided a local loop at the endsites and also provided the capability to hand off at 1.544 Mbps (T1).

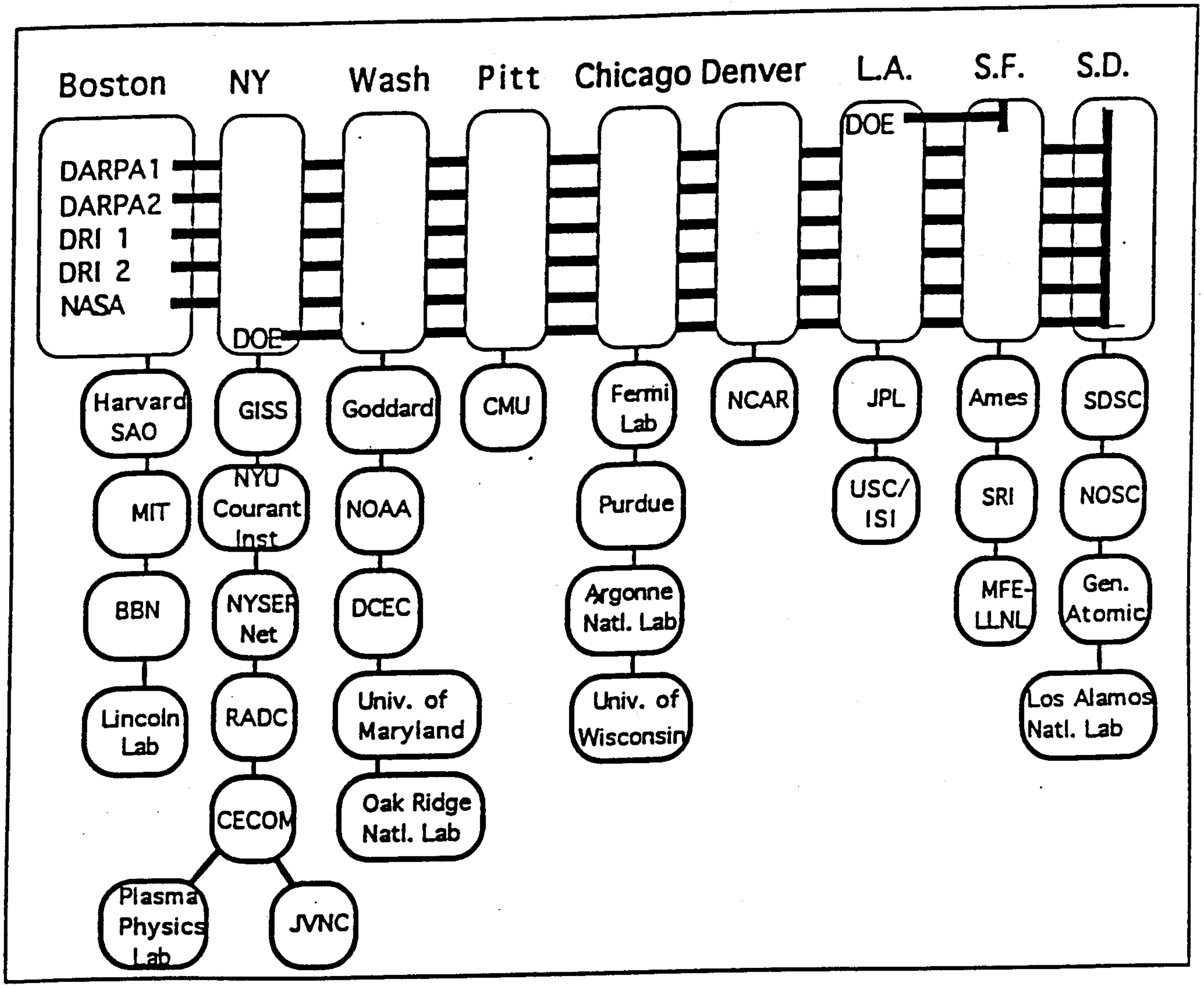


Figure 2.
Initial Topology of National Networking Testbed

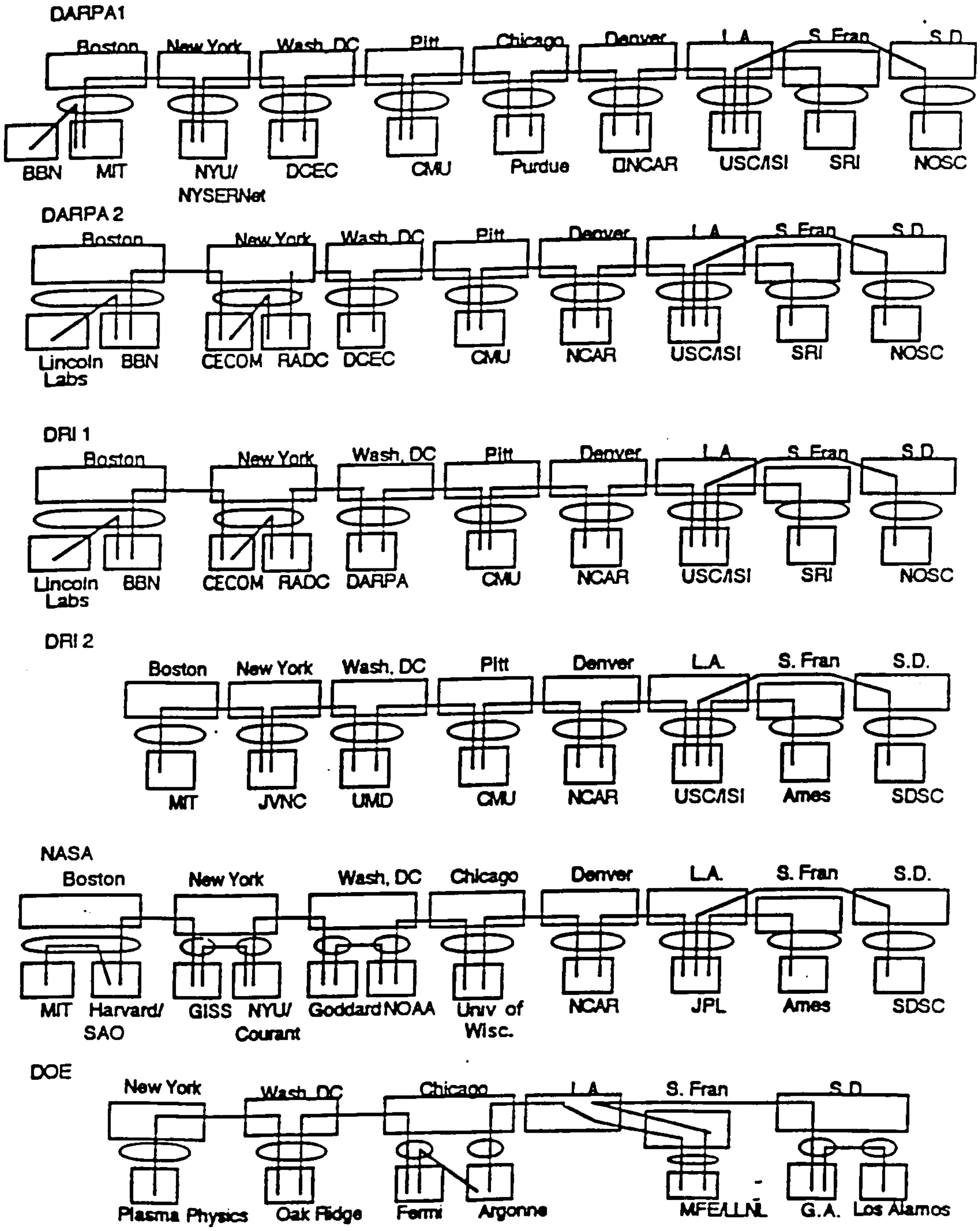


Figure 3.
Detailed Topology with Routers at End-Sites Only

2.2.3 Services of the NNT

In developing the NNT topology, PSI and NYSERNet, Inc. provided the networking services of the National Networking Testbed as necessary or as requested. Those services included the following key components:

- Provision of POPs in key cities along the DS3 route (Boston, New York, Washington, Pittsburgh, Chicago, San Francisco, Los Angeles). Expansion of POPs as equipment requirements exceeded space in Washington. Added new POPs in Mobile, Kirkland, and Ames (NASA).

- Provided routine and emergency access for contractors of the Defense Advanced Research Projects Agency to enter the POPs at any time. Thus, they could deliver and remove equipment on an as-needed basis.

- Specified, accepted, installed, and assisted in the management of government furnished equipment (GFE). The GFE included equipment racks, circuit termination equipment, and circuit test equipment.

- Designed and installed backbone and tail circuits on a routine schedule and in emergencies. The circuit provisioning procedures specified by DARPA were used.

- Provided emergency equipment and repair support personnel during equipment failures. Provided stand-by technicians and engineers during key application demonstration projects.

- Provided continuous monitoring of operations of all network circuits and equipment under the purview of the NNT scope of work.

- Detected and diagnosed trouble; contacted relevant NNT sites and networks; resolved trouble with the relevant telephone companies.

- Provided network engineering for NNT-related circuit design and installation efforts in conjunction with the telephone companies. This approach took advantage of an interface to the Extended Super Frame and other in-band and out-band monitoring facilities of the telephone companies with SNMP tools used for the National Networking Testbed.

- Provided network engineering support for network equipment attached to the National Networking Testbed and integrated computer systems used for switching, routing, circuit termination, reliability enhancements, and security. Worked with DARPA contractors to troubleshoot the installation and maintenance of these facilities.

- Provided engineering change proposals and contract amendment support. Assisted in negotiations with other suppliers, as needed, to meet the changing requirements of the Defense Advanced Research Projects Agency throughout the contract period. The changes included increasing the number of sites that can be connected to the National Networking Testbed, extending the duration of the contract, decreasing the minimum number of backbone circuits on a supplier contract, and building and maintaining engineering staff relations with DARPA contractors.

- Provided engineering, operation, and maintenance liaison for the Defense Advanced Research Projects Agency in activities related to the National Networking Testbed. The same services were also provided to the National Science Foundation (NSF), DOE's Livermore Laboratory, NASA's Ames Research Center, and several MILDEP research and operational facilities located in the Continental United States (CONUS).

- Met regularly with management and technical leadership of the Defense Advanced Research Projects Agency and Rome Laboratory to match the development schedule and design of the National Networking Testbed to Government requirements.

- At the completion of the testing period, provided for the smooth termination of NNT connections and reintegrated the sites into other network systems.

- Provided monthly activity and status reports of circuit installations, de-installations, POP construction, site problems, etc.

2.2.4 Network Operations Centers

PSI and NYSERNet, Inc. expanded the Network Operations Center in Troy, New York. This Center was integrated with the National Network Monitoring Center of the Williams Telecommunications Group in Tulsa, Oklahoma and the Network Operating Centers for the various components of the National Networking Testbed. The Troy Network Operations Center was then able to provide continuous, 24-hr/day surveillance and trouble shooting. It could provide recovery assistance to the associated networks.

A separate 9.6 kbps dialup connection was installed from each National Networking Testbed multiplexor site to each POP to provide a secondary method for monitoring network problems. This line provided an independent source of information on the status of the network for all city-pair lines and multiplexers. Surveillance included line outages, transient line, and multiplexor problems. The same information that was available at the Network Operations Center of the Williams Telecommunications Group in Tulsa was available at the PSI-NYSERNet, Inc. Network Operations Center in Troy.

The National Networking Testbed was demonstrated for four and one half years. During that time, nearly one hundred sites were connected to six different network backbones. A listing of the networks provided, the name of the site connected, and dates of connection and de-commissioning are presented in Table 1. Topology diagrams depicting the NNT supporting six different networks (TWBNet, CORNETT, ESNET, DARTNet, Los Nettos, and Arpanet) in May 1992 are included in Appendix A.

Table 1. Site Connections Provided Under the National Networking Testbed

Origination Site	Termination Site	Activate Date	Deactivation Date
Arpanet (DARPA)			
	NISCTroy	1/1/90	4/6/90
	BBN	1/1/90	4/6/90
	BBN	1/1/90	6/1/90
	DCEC	1/1/90	6/1/90
	DCEC	1/1/90	4/9/90
	CMU	1/1/90	2/28/91
	CMU	1/1/90	1/21/90
	Ft. Ben	1/1/90	4/9/90
	Ft. Ben	1/1/90	4/9/90
	ISI	1/1/90	4/9/90
	ISI	1/1/90	4/3/90
	SRI	1/1/90	4/3/90
LOS NETTOS (DARPA)			
LA-POP	ISI	1/1/90	2/23/93
SD-POP	NOSC	5/23/90	2/23/93
CORNETT (DOE)			
CHI-POP	ANL	1/1/90	3/5/91
LA-POP	LANL	1/1/90	1/22/91
LA-POP	Caltech	1/1/90	2/28/91
decomm.	SRI	1/1/90	3/15/90
KIRT-POP	LANL	1/22/91	2/28/91

Table 1. Site Connections Provided Under the National Networking Testbed (cont.)

Origination Site	Termination Site	Activate Date	Deactivation Date
ESNET (DOE)			
BOS-POP	MIT	1/1/90	7/1/91
BOS-POP	MIT	1/1/90	7/1/91
NYC-POP	BNL	1/1/90	7/1/91
NYC-POP	NYU	1/1/90	7/1/91
NYC-POP	PPPL	1/1/90	7/1/91
NYC-POP	PPPL	1/1/90	7/1/91
SF-POP	LLNL	1/1/90	7/1/91
CHI-POP	LBL	1/1/90	7/1/91
CHI-POP	Fermi	1/1/90	7/1/91
CHI-POP	Fermi	1/1/90	7/1/91
Terrestrial Wide Band (DARPA)			
BOS-POP	BBN	1/1/90	2/26/93
NYC-POP	CECOM/Ft.Mnth	1/1/90	3/17/92
NYC-POP	CECOM/Ft.Mnth	1/29/92	2/26/93
NYC-POP	RADC	1/1/90	3/25/92
NYC-POP	RADC	2/25/92	2/26/93
DC-POP	Ft. Rucker	1/1/90	2/8/91
DC-POP	DARPA	1/1/90	7/19/91
DC-POP	DARPA	6/21/91	3/22/93
DC-POP	NRL	1/1/90	2/11/93
DC-POP	SIMNET	1/1/90	9/3/91
DC-POP	SIMNET	8/13/91	9/18/92
PITT-POP	CMU	1/1/90	4/16/92
CHI-POP	NCSA	1/1/90	3/12/91
CHI-POP	Ft. Knox	1/1/90	3/22/93
CHI-POP	Ft. Leaven	1/1/90	3/27/91
CHI-POP	Ft. Leaven	1/27/92	3/22/93
CHI-POP	AFIT	7/28/92	3/22/93
CHI-POP	TRANSCOM	8/14/92	3/22/93
LA-POP	ISI	1/1/90	3/22/93
SF-POP	SRI	1/1/90	3/1/91
LA/SFPOP	SFO/SRI	7/22/92	3/22/93
Ames	RIACS	7/20/90	3/2/93
RIACS	Ames	7/20/90	3/22/93
LA-POP	Ames	7/9/90	3/22/93
DC-POP	FIX/SURA	7/3/90	3/22/93
LA-POP	NOSC	8/2/90	8/22/92
LA-POP	NOSC	4/16/92	3/22/93
DC-POP	Ft.Belvr-ETL	10/1/90	12/23/92
DC-POP	TEC	11/17/92	3/22/93
DC-POP	BBN	9/10/90	3/25/91
Kirt-POP	Mobile POP (2/23/93)	9/11/90	7/14/92
LA-POP	Kirtland POP	12/12/90	9/16/92
MOB-POP	Ft. Rucker	9/12/90	7/14/92
MOB-POP	Hurlburt	3/27/92	7/8/92

Table 1. Site Connections Provided Under the National Networking Testbed (cont.)

Origination Site	Termination Site	Activate Date	Deactivation Date
Terrestrial Wide Band (DARPA)			
KIRT-POP	LANL	12/12/90	9/4/92
DC-POP	Quantico	4/16/92	3/22/93
DC-POP	Mitre	3/16/92	9/19/92
DC-POP	Mitre	5/26/ 92	3/22/ 93
DC-POP	DCA/DCEC	5/27/92	3/22/93
DC-POP	IDA	4/27/92	3/22/93
DC-POP	USAF/Pentagon	4/27/92	3/22/93
DC-POP	NATC,Patuxent	10/19/92	3/22/93
DC-POP	AFSC(Norfolk)	9/ 18/ 92	3/22/93
DC-POP	Army WC(PA)	8/17/92	3/22/93
DC-POP	BRL (Aberdeen)	9/18/92	3/22/93
DC-POP	Natl.WC,McNair	10/1/92	3/22/93
DC-POP	CECOM/Ft.Mnth	12/17/92	3/22/93
DC-POP	BBN	12/17/92	3/22/93
DC-POP	NRL	2/8/93	3/22/93
DC-POP	NSWC,Cardrock	9/22/92	3/22/93
DC-POP	Gunter	7/31/92	8/28/92
LA-POP	Korea (256)	5/15/92	12/23/92
LA-POP	Korea (512)	8/6/92	3/22/93
LA-POP	NTC(Ft. Irwin)	7/22/92	3/22/93
LA-POP	PMTC(Pt Mugu)	12/28/92	3/22/93
LA-POP	KAFB(Kirtland)	7/22/92	9/20/92
LA-POP	CSDRF (Ames)	10/7/92	[1/8/93]
DARTNet (DARPA)			
SF-POP	LBL	4/3/90	10/19/92
Ames	Xerox PARC	5/16/90	10/19/92
Ames	SRI	5/16/90	10/19/92
SRI	Ames	7/16/90	10/19/92
Xerox PARC	Ames	7/26/90	10/19/92
SC-POP	Ames	7/26/90	10/ 19/92
LA-POP	ISI	4/3/90	10/19/92
DC-POP	UDEL	1/1/90	10/19/92
BOS-POP	BBN	6/1/90	10/19/92

2.2.5 NNT Networks and Users

a. DARPA made significant use of the NNT through four preexisting network systems that were reengineered and made part of the NNT for testing purposes. These networks, which previously relied on either leased lines or satellite facilities, included the following:

Terrestrial WideBand Network (TWBnet), a high speed fiber optic replacement for the earlier WideBand network that relied on satellites. TWBnet uses Bolt, Beranek, and Newman (BBN) routers and related equipment.

DARTNet, a T1 network backbone that connected specific research and development sites in order to test component systems, such as networking equipment and applications from other sources.

Los Nettos, an independent metropolitan area network that supports ISI projects.

Arpanet, the first Internet protocol (IP) network. Arpanet was shifted almost entirely to the National Networking Testbed's speed of 1.544 Mbps (T1) during the first 18 months of the Wide Area Networking R&D contract.

b. The U.S. Department of Energy used the National Networking Testbed for two network systems, ESNET and CORNETT. ESNET is DOE's primary networking system. CORNETT was an experimental network. It supported various DOE research applications.

   DOE withdrew from the National Networking Testbed after approximately two years. The Department said that it required direct control of all components of its network design, system integration, and operation.

c. NASA did not take advantage of the National Networking Testbed. The agency said that its networking systems were mission-critical, and NASA, therefore, had to have total end-to-end control. The agency's need for control transcended cost savings or enhanced reliability from access to POPs.

A NASA system was discussed, but it was never implemented. The system was developed for the Numerical Aerodynamic Simulation network (NASNET) and for the NASA Science Internet (NCI), the general purpose network for NASA research activities. The NASA Ames Research Center (ARC) operates NASNET and manages NSI.

2.2.6 NNT Transition

During the last year of the contract Rome Laboratory, based on DARPA's request, directed NYSERNet, Inc. to incorporate enhancements to the TWBnet and DARTNet, the two remaining networks managed under the NNT. DARPA interest in further refinement of these networks stemmed both from equipment problems experienced with DARTNet and a Memorandum of Agreement (MOA) between DARPA and DISA concerning transition of TWBnet. The enhancements to

DARTNet, which basically were one generation advanced from the TWBnet technology, were necessary to correct technical equipment difficulties that had delayed the completion of certain experiments and to refine the DARTNet technology before it was deployed throughout the Internet. The DARPA/DISA MOA agreed to transfer a version of the TWBnet technology, and operational control, to DISA in the March 1993 time frame for evolvement into DISA's planned Defense Simulation Internet (DSI).

In September 1992 NYSERNet, Inc. was directed to proceed with the termination of all lines and support for DARTNet, to be effective on or after October 1992. DARPA planned to utilize residue equipment from DARTNet to build DARTNet II. With DARTNet terminated, the TWBnet remained as the sole national R&D network supported by the NNT.

On December 28, 1993 DARPA hosted a meeting to discuss transition of the TWBnet to DISA, to include the planned disconnect of existing circuits and the removal of POP hardware. The objective was to have the transition completed by the end of March 1993. DARPA would trigger disconnect requests and order new circuits when DISA's new backbone was installed. On March 22, 1993 all remaining TWBnet circuits subcontracted for under the NYSERNet, Inc. contract were disconnected. Based on previous orders placed by DARPA with the Defense Commercial Communications Office (DECCO), many of these circuits were reconnected to comprise the initial configuration of DISA's new DSI. Selected TWBnet data communication circuit termination equipment was left in place to accommodate the cutover to DSI. Accountability of this equipment, which was considered residue property under the NYSERNet contract, was subsequently transferred to the DSI support contract. Accountability of other residue equipment under the NYSERNet contract was transferred to the DARTNET II support contract and Rome Laboratory.

Section 3
SNMP/X.500 Directory Software Development

This section describes the research and development effort supporting advanced internetworking. This work, which was performed by PSI, centered on the following two areas:

- Network management client and server systems software and international standards which improve the end-to-end performance and reliability of the Internet by enabling detailed remote monitoring, trouble detection and correction;

- Applications server and client software and international standards implementations which improve the usability of the Internet by providing global users with remotely accessed and controllable "directory information" systems.

Efforts included national coordination and leadership at the Internet Engineering Task Force (IETF), the North American Directory Forum (NADF) and other relevant technical committees. This work involved technology transfer coordination with national and international commercial, academic and government organizations on the Simple Network Management Protocol (SNMP), International Standards Organization Development Environment (ISODE) and X.500 Directory software development. Results required an ongoing general consensus building in the multiple areas of technology direction covered by this effort.

Work by PSI has included significant demonstration of research prototypes, with some prototypes being successfully moved from the public domain R&D level to commercial products (in SNMP), with some remaining in the public domain environments (X.500), as well as extensive operational testing and use throughout the global Internet. This software development effort made use of the extensive capabilities afforded by the National Networking Testbed (NNT).

3.1 Principal Areas of Effort

The principal areas of effort under the SNMP/X.500 Directory Software Development task included the following:

- SNMP Working Group Participation
- ISODE Development, Support, Futures
- Operation of the White Pages Project (WPP)
- Field Operational X.500 (FOX) Project Cooperation
- Explore Content and Acceptance Extensions to WPP
- Lightweight Directory Access Protocol (LDAP) Development

- SNMP Extensions to Fast Packet/Cell Relay
- Documents and Technical Reports
- Software Development and Availability
- Other Activities

Work performed in the above areas is summarized in the following subparagraphs.

3.1.1 SNMP Working Group Participation

A PSI staff member (Dr. Marshall Rose), chaired the Simple Network Management Protocol (SNMP) Working Group (WG) of the Internet Engineering Task Force (IETF) during early 1991. The IETF is the primary group of the Internet Activities Board (IAB) under which DARPA and other federal agencies coordinate nationwide internetworking activities which are part of the worldwide Internet. Dr. Rose left PSI's employ during 1991, and continued to make his contributions to the SNMP-WG, with PSI's full participation in the group's deliberations and work.

SNMP was named the primary management tool for TCP/IP networks by the IAB in 1992. PSI has maintained a consistent approach to review the changing requirements of SNMP's supporting data structures and information databases, specifically its Management Information Base (MIB-II, and many more focused MIBs including T1, T3, Ethernet, FDDI, fr, etc.

3.1.2 ISODE Development, Support, Futures

The ISO Development Environment, (ISODE) was significantly extended under this contract so that the European and US networking communities would be able to interoperate more smoothly. This effort included significant work in applications development and ISODE documentation.

In 1991, PSI participated in the initial planning of a not-for-profit international consortium to provide a permanent "home" for this critical software. Called the ISODE Consortium, PSI was a co-founder in the initial formation, which is dedicated to pursuing this technology in a cooperative fashion as initially developed under the leadership of Dr. Marshall Rose. The Consortium selected Steve Hardcastle-Kille as president, and has established its headquarters at ISODE Consortium, European Office, P.O. Box 505, London, SW11 lDX, UK, phone +44-71-223-4062; with an office at ISODE Consortium, c/o MCC, P.O. Box 200195, Austin, TX78720, USA, +1-(512)-338-3340.

3.1.3 Operation of the White Pages Project

The White Pages Project (WPP) was expanded under this contract from its 1988 beginning 6 member organizations with several thousand entries to the current 180 organizations with between 1 and 2 million entries. The WPP runs on both TCP/IP and OSI and is the largest operating directory service on the global Internet. The

WPP is being used as a model for the new (spring 1993) National Science Foundation "INTERNIC" contract award to AT&T, which will be constructing an X.500 whitepages and yellow pages directory service for the NSF.

A list of the current participants is included in Appendix B.

3.1.4 FOX Project Cooperation

During the period of this contract, DARPA funded a joint effort led by the Information Sciences Institute (ISI) at the University of Southern California, with participation by PSI, SRI, and Merit called the Fielding Operational X.500 (FOX) project. The primary objective of the FOX project was to unite separate but cooperating directory service efforts across the Internet. FOX provided a platform for WHOIS entries to migrate to X.500, and an experiment in interoperability and standards development among the leading contenders in directory service standards technology, including Quipu, National Institute of Standards and Technology (NIST) and Digital Equipment Corporation's (Digital Network Architecture) Name Service (DNANS) implementation of X.500. PSI provided direct coordination between the FOX project and the efforts under this contract.

3.1.5 Explore Content and Acceptance Extensions to WPP

In an effort to determine how far X.500 directory technology might eventually extend to support the Internet, PSI explored adding both content fields and widening the availability of X.500 access. To begin, new types of information content were added to the X.500 White Pages Project directory pilot. The X.500 standard is silent about usage and policy, and PSI explored content types beyond organizations and people. An X.500 client was created for the DOS operating system, since simple PC clients address a much broader audience for the technology.

To share what was learned in this process, PSI supported the North American Directory Forum (NADF) X.500 efforts and participated in the Internet Engineering Task Force (IETF) OSI-DS group. The experience gained from both the new content and DOS client access is directly applicable to the implicit policy charters of NADF, the IETF OSI-DS working group, and the ISODE Consortium.

3.1.6 LDAP Protocol Development

The Lightweight Directory Access Protocol (LDAP), was developed for use by PC client software, and it is currently being considered for Proposed Standard status by the IETF. The X.500 standard defines a Directory Access Protocol (DAP) which operates in an OSI framework. The most common implementation is QUIPU, which uses the ISODE to field X.500 within the TCP/IP community. The DAP is a high overhead protocol and is unsuitable for the PC class machines that are quickly comprising the bulk of Internet clients. The LDAP is designed to provide low overhead directory access for PC based clients.

3.1.7 SNMP Extensions to Fast Packet/Cell Relay

In early 1992, PSI described to Rome Lab and DARPA its plans to extend the Simple Network Management Protocol (SNMP) to the newest fast-packet and cell relay based offerings by telephone companies. These new services would be the most likely layer two medium for future Internet systems, and they are currently not accessible by any SNMP MIB. Among those which seemed most interesting were FrameRelay, ATM and SMDS.

PSI worked with a frame relay vendor to apply SNMP to frame relay network management and created an SNMP based IP traceroute tool, called 'fr', that also exposes underlying frame relay routing. An important tool in IP network debugging is a program called 'traceroute'. Traceroute is used to examine the route used by packets traveling from one host to another host. This knowledge is used to diagnose routing problems as well as route failures, i.e. the inability of one router to pass packets to the next router. Many route failures are caused by a failure in the underlying medium. For simple media, such as leased T1 lines or local ethernets this diagnosis is straightforward. However, for compound media like a frame relay path, the diagnosis is much more complicated. The 'fr' tool exposes the simple media within a frame relay path so that straightforward route failure diagnosis is maintained.

3.1.8 Internet Draft Documents

Wengyik Yeong of PSI participated in the development of the following Internet Draft documents:

- Internet Draft Application Monitoring MIB. This document defines a MIB for monitoring OSI applications, specifically X.500 Directory Service Agents (DSAs) and X.400 Message Transfer Agents (MTAs)

The ability to observe the operation of X.500 DSA is critical to providing reliable directory services in an internet environment. Using this proposed MIB the network manager can observe the loading and failure rate of compliant X.500 DSAs (see Appendix C.)

- Internet Draft "The String Representation of Standard Attribute Syntaxes". This document defines requirements for lightweight directory protocol attribute syntax encodings. (See Appendix D)

- Internet Draft "Lightweight Directory Access Protocol". This specification is currently undergoing review for advancement to Proposed Standard status in the IETF (See Appendix E)

3.1.9 Software Development and Availability

- "A PC front-end for the Directory Assistance Service (RFC1202)" [1]. This is a DOS command line program that uses the Directory Assistance Service (DAS) over FTP Software's PC/TCP kernel. A Windows version was explored but was deferred because of the immature state of TCP/IP for Windows at the time.

The DAS is an early attempt at a low overhead directory protocol for PC class machines. This program uses the Directory Assistance Protocol (not to be confused with the Directory Access Protocol). This program provides a command line interface similar to 'fred'.

- Software to load Domain Name System (DNS) resource records (RRs) into the X.500 trees. DNS information that has been loaded can be seen under

{ o=Internet, dn=arpa }
{ o=Internet, dn=net }
{ o=Internet, dn=com }
etc ...

This software produces X.500 objects as described in RFC1279.

WHERE: Anonymous ftp from ftp.psi.com as wp/dns2edb.tar.Z.

A goal of this effort was to place pointers from DNS entries to relevant White Pages entries. For example, a query for 'user@psi.com' could return the postal address and phone number of that user. These links were installed for PSI, NYSERNet and XTel but wider acceptance was hampered by the effort required to identify the correct pointers for each participating organization.

- Modified the 'fred' interface to use DNS information available in the X.500 Directory Information Tree (DIT) in resolving White Pages queries.

WHERE: This software was never released because the specification for representing DNS information in the Directory changed. However the implementation experience was presented to the IETF osi-ds group.

These software changes provided a user interface for the DNS mechanism described in item 2. It detects the usage of a domain address (i.e. user@foo.com) and uses the DNS information in the directory to find the White Pages entry for the user.

- Software to produce DSA configuration files that conform to the NADF specification. This software is required for participation in the White Pages Project.

WHERE: Anonymous ftp from ftp.psi.com as wp/pilot/wpp-addon.tar.Z.

The NADF has specified state and county level directory entries underneath the national entry. The White Pages DSAs were conformed to this new directory structure using this software.

- Software to generate knowledge and naming updates from the DIT for use in the NADF CAN (Central Administration for the NADF).

WHERE: This software has since been superseded by software written by Marshall Rose, and is therefore no longer being distributed.

X.500 specifies a master/slave data replication mechanism. This structure is inadequate in a competitive directory service environment. The NADF has specified a master/master replication mechanism which this software implements. Using this mechanism DSAs maintained by separate service providers can keep each other up to date.

Modified the 'fred' interface to accommodate the NADF naming scheme.

WHERE: This software is available as part of the (publicly available) ISODE distribution.

Prior to the NADF naming scheme, all White Pages organizations were of national standing and were located under their respective national entry. These changes modified the User Friendly Naming (UFN) algorithm to account for organizations not of national standing.

- Started an implementation of the Lightweight Directory Browsing Protocol (LDBP).

WHERE: This effort was never completed when LDBP was superseded by LDAP.

The Directory Access Protocol (DAP) is a very general protocol. Experience with the White Pages demonstrated that this generality imposes a large processing overhead particularly when handling DSA referrals. The LDBP was designed to provide a low overhead search and read capability for White Pages clients on small machines.

- Currently writing an implementation of the Lightweight Directory Access Protocol (LDAP).

WHERE: Not completed.

LDAP has been developed along the same lines as LDBP but represents the efforts of a larger community. In addition to search functionality, a modify/add/delete capability is also included to support decentralized directory administration. Abstract Syntax Notation 1 (ASN.1) is used for protocol data unit (PDU) encoding.

- Software using SNMP to trace IP routes and expose underlying frame relay routes where they exist.

WHERE: Anonymous ftp from ftp.psi.com as snmp/fr

The experience gained from this tool points out a need for standard MIB information for managing multi-tiered networks (e.g. running an IP network on top of a frame relay network). The standard MIB generally assumes unintelligent IP interfaces. However, if the interface is intelligent and supports SNMP queries, there is no standard way to query a router to determine how to query that interface.

For example: PSINet uses Cascade frame relay switches connected to Cisco routers. While following an IP route through the Cisco routers using the MIB variable ipRouteNextHop, the type of interface is determined by the MIB variable ifType. If the interface is of type frameRelay then the route taken by the frame relay packets to the next Cisco router must be examined. To do this the following information is required about the Cascade switch connected to the Cisco: its IP address, the Data Link Connection Identifier (DLCI) used and the logical port used within the Cascade switch. There is absolutely no MIB support for this information in either the standard MIB or the Cisco extended MIB.

The current procedure telnets to the Cisco router and issues commands to identify the DLCI used at the interface. An external table is then consulted to identify which frame relay switch is connected to this router. This is done by partially matching the sysName MIB variables. The private MIB of the frame relay switch is then walked to find the correct DLCI entry. Note that the same DLCI can be used by more than one frame relay interface on the frame relay switch. In order to identify the correct DLCI an administrative policy specifying unique DLCIs within a switch must be enforced.

Clearly this methodology does not scale well in a large and multi-vendor internet. Standard MIB mechanisms are required to traverse intelligent interfaces without resorting to ad hoc methods.

3.1.10 Other Activities

- Presentations at Interop '91 on White Pages Pilot Project and X.500.
- Participation in the development of the NADF agreements on naming and knowledge sharing.
- Participation in the IETF osi-ds group.

3.2 Activity Reports

The following monthly summaries chronologically describe the activities under the SNMP/X.500 Directory Software Development task.

April, 1991

Marshall Rose completed the Final Questions and Answers for three Network Management RFCs:

1) Concise - a technique for writing MIBs for use with SNMP which results in no loss of semantic content but achieves a 46% reduction in MIB size.
2) Traps - a convention used by vendors when defining traps specific to their agents.
3) MIB-II - The new Network Management Information Base. These RFCs are now being prepared to be issued formally.

Work was also accomplished on a new X.500 Directory System Agent - Directory Information Tree and Quality of Service (DSA/DIT QoS) attributes and better User-Friendly Naming (UFN) diagnostics.

Rose and others also reviewed more technical papers for SIGCOMM '91 concerning Network Management issues.

A new White Pages Project (WPP) site was added, #74, Sun Microsystems Incorporated, Corporate, with Barry Holroyd as representative, berries@eng.sun.com 415-336-2949.

May, 1991

During the week of April 1, 1991, Dr. Marshall Rose attended the IFIP 6.6 Symposium in Washington, DC. This conference brought together many members and non-members of the International Federation of Information Processing who are interested in X.500 and other international standards.

During April and the early part of May, continuing interaction with users of ISODE resulted in the identification of additional software bugs. Many of these were successfully fixed, with the resulting fixes made available in the code which is available by ftp from PSI.

A public domain software package for the Personal Computer environment, providing a Directory User Agent (DUA) for the White Pages X.500 pilot has been designed and is in the process of being implemented.

This work was accomplished by Wengyik Yeong, PSI Staff Scientist, and is in alpha test within PSI. This work is an example of prototype software built on software standards. It is being implemented in the C language, using the PC/TCP networking kernel from FTP Software, Inc. The intent is to have this DUA operate on most DOS based PC's, allowing individuals to access all of the participating DUA's on the Internet. When complete, this software will be put in the public domain by PSI and distributed free via ftp.

June, 1991

Beta testing was performed on ISODE beta-release 6.9. Experimental versions of existing DSA and DUA software were tested. Testing was also performed to ensure that version 6.9 is backwards-compatible with version 6.8.

Considerable effort was expended in the past month to increase the reliability of operational X.500 service. To this end, several procedural and software modifications were made:

1. watchdog scripts, that ensure that processes related to the provision of X.500 service continue to run, were written and installed. These scripts will be further enhanced in the future to detect more error conditions that cause denial of service.
2. some shuffling of DSA functionality among machines was performed to better distribute the demands placed on various machines by X.500 service.
3. an enhanced version of the ISODE software was installed on all machines to fix version skew problems and further enhance reliability.
4. a systematic schedule of monitoring the reliability of all PSI machines that contribute to X.500 service was begun.

Testing was performed, using some of the data in the o=Performance Systems International subtree, in preparation for alignment with the Cooperation for OSI Networking in Europe (COSINE) /Internet Naming Architecture.

Final testing of BETA PC/MSDOS DUA nearly completed - this is a layered application over FTP's PC/TCP Kernel. Anonymous FTP release of the binary is slated for 1st week of July.

July, 1991

Work in July saw continuing efforts begun several months ago to bring the X.500 service into full operational mode. In an effort to improve general reliability of White Pages service, a number of scripts were written to automate the gathering of DSA reliability statistics and to detect the failure and/or absence of X.500 quipu DSAs. All these scripts are currently being tested, and it is anticipated that the failure detection script will be fed back into future ISODE distributions.

In anticipation of the release of ISODE 7.0 [NOTE: ISODE 7.0 has now been released], some testing was performed to determine the upgrade steps required for White Pages Project participants to transition from existing software and data to ISODE 7.0.

The 6th meeting of the North American Directory Forum was attended, at which the Forum was continuing to develop plans to help develop an international directory.

A beta version of an MSDOS-based front end to the White Pages, 'pcwp' was released. This application was subsequently modified based on user input, and released again. More modifications will be performed based on further user input, and it is anticipated that the application will be released yet again in the middle of August.

Several draft RFC's, IDEAS, and NADF documents concerning X.500 and its applications were reviewed and suggestions for modifications made. Additional reporting applications were being developed and written to report on DSA's.

PSI coordinated this X.500/ISODE/SNMP work with the FOX project, which included writing scripts to automate the update of information in the o=Internet@cn=RFC Documents tree and the loading of uumap information into the DIT.

As of July 24, 1991, there were 77 organizations participating in the White Pages in the United States of America. New organizations added this past month are
University of Florida
The Mitre Corporation
Organizations deleted from the US arc this past month are:
University of Minnesota

August, 1991
Work in July and August to bring the X.500 service into full operational mode has proven successful. In operational tests with participating organizations, reliability of the White Pages service has improved.

Modifications were made to the 'pcwp', MSDOS front end to the White Pages, and a second beta version was released at the end of this month. It is available for anonymous ftp from uu.psi.com in wp/pcwp.exe. We expect this application to become one of the primary means for users to access the prototype system. More modifications will be performed based on further user input, and it is anticipated that the application will be released yet again later this fall.

Work is progressing on the design and implementation of a full-screen front-end to the White Pages for MSDOS.

PSI coordinated this X.500/ISODE/SNMP work with the FOX project, which included writing scripts to automate the update of information in the o=Internet@cn=RFC Documents tree and the loading of uumap information into the DIT.

A script was written to retrieve listings of the contents of archives available for anonymous ftp in preparation for extensions to the x5ftp retrieval application.

A proposal for the organization of the US DMD was prepared, based on planning and testing (in the testbed, below) performed during the month.

Testbeds were set up to perform some testing on X.500-related proposals. In the past month, testing was performed on the IETF OSI-DS group's recommendations on DSA naming, and to test a proposed plan for the organization of the US DMD. Some minor testing was also performed to verify that changes made to the US public naming scheme in the last NADF meeting did not cause any problems.

As of today, there are 80 organizations participating in White Pages in the US. New organizations added to the pilot this past month are:

The Mitre Corporation
Vitalink Communications Corporation
Duke University
Hughes Aircraft Co.

Organizations deleted from the US arc in the past month are:
University of Alaska at Fairbanks

September, 1991
Operational status of the Pilot White Pages service remained smooth throughout the month, as a result of the software and hardware improvements made earlier in the year.

Experience and feedback from users on the new "pcwp" MSDOS front end to White Pages has begun after its release in August. Code development will continue as user feedback is collected.

Work is progressing on the design and implementation of a full-screen front-end to the White Pages for MSDOS.

PSI coordinated this X.500 work with the FOX project, which included:

a) A program to automatically load RFC information into X.500 was modified to include RFC numbers as a search key so as to allow the 'x5ftp' program to search by RFC numbers, in addition to other existing search keys. And,
b) In preparation for alignment of the White Pages Pilot Project with North American Directory Forum (NADF) naming scheme, the 'usconfig' program was written. The 'usconfig' program performs the same function as the existing 'dsaconfig' program, except that it has knowledge of the NADF Naming recommendations, and produces configurations specific to the U.S.

A new draft on the naming of DSAs, now released as the Internet Draft draft-ietf-osids-dsanaming-01.txt was reviewed, and comments sent to the author.

The manual "PSI White Pages Project: Administrator's Guide"[2] was updated to reflect the upgrade to ISODE 7.0, and the upcoming transition to the NADF's recommendations on naming.

One new organization was added to the pilot this past month:
Dover Beach Consulting, Inc.

October, 1991

Operational status of the X.500 White Pages Pilot service remained stable throughout October, due to improvements made earlier in 1991.

User experience on the "pcwp" MSDOS front end to White Pages has identified where additional work will be focussed. Released in August, code development will continue throughout the fall, while additional user feedback is collected.

Work is progressing on the design and implementation of a full-screen front-end to the White Pages for MSDOS.

PSI coordinated this X.500 work with the FOX project, contributing a new Draft RFC Information Document to the Networking Working Group. This work is an extension to the program to automatically load RFC information into X.500 called the 'x5ftp' program, which allows searches by RFC numbers and other existing search keys. Currently entitled "Representing Public Archives in X.500"[3], the draft document is attached. This document was first submitted 23 Oct 91, by Wengyik Yeong. This is the schema for the next phase of 'x5ftp' development. It will be sent to the general osi-ds list after the FOX group has provided comments.

November, 1991

Operational status of the X.500 White Pages Pilot service remained stable throughout November, due to improvements made earlier in 1991.

Work is progressing on the design and implementation of a full-screen front-end to the White Pages for MSDOS.

The manager of the c=US arc in the global DIT participated in cleaning up the root of the DIT. To this end, the l=North America node was deleted from the DIT as per the agreements reached in the last meeting of the IETF OSI-DS group. In addition, the manager of the c=US arc, in cooperation with the managers of other national arcs worked to ensure consistent replication of national entries among top-level DSAs in order to enhance the operational stability of various Directory pilots.

A fix to the 'fred' program was installed to circumvent an assumption in the search algorithm used by 'fred'. The fix will be returned to the maintainers of the quipu software once it is determined that the problem the fix is supposed to resolve no longer exists.

In preparation for participation in the NADF Experimental Directory pilot, some discussions were held with other prospective participants to coordinate various details involved in the setup of the pilot.

A decision was made to rename the existing 'x5ftp' application to 'x5rfc', to more accurately reflect its function. A new application, called 'x5ftp', that retrieves files from anonymous ftp archives was written and completed. The document describing the schema underlying the (new) 'x5ftp' application was submitted to the OSI-DS group for consideration.

The transition of the PSI White Pages Pilot Project to the NADF naming scheme continues.

New organizations added to the pilot this past month are:
University of Oregon

December, 1991

In preparation for participation in the NADF Experimental Pilot, the NADF KAN software was modified and installed to coexist with the existing White Pages Pilot Project environment. This project will attempt to make all North American efforts in network directory services compatible in some primary applications.

Work was begun on software to make some of the information from the White Pages Pilot Project available in the NADF Experimental Pilot.

Use of the White Pages servers are increasing, with over 500,000 entries at this date. A sublisting will be made available in a separate report. Due to increasing load on the Fruit Bat DSA, a new DSA, c=US@cn=Horned Frog was installed on the WPP service machine wpl.psi.net to relieve the Fruit Bat DSA of its duties as a White Pages service DSA. The Fruit Bat DSA will now function solely as a backup for Alpaca, the c=US MASTER.

In anticipation of future work involving the Domain Name Server (DNS) and X.500, two new DSAs, c=US@cn=Swamp Fish and c=US@cn=Hatchet Fish were created. Authority for the toplevel domainComponents for .us, .org, .net, .mil, .gov, .edu, .com and .bitnet were transferred to the Swamp Fish DSA, with the Hatchet Fish DSA serving as a backup.

New organizations added to the pilot this month are:
Stanford Linear Accelerator Center
University of Mississippi

The following list is the sum of all national entries available in the international White Pages project. The total has exceeded 500,000 entries and remains the largest, most complete list in existence for TCP/IP and OSI based network directory.

Node	DSAs	Entries	(DSAs unavailable)
AT	3	1516	2
AU	15	29246	2
BE	1	0	1

CA	19	16466	9
CH	7	25202	2
Cosine	3	0	1
DE	21	2653	1
DK	1	688	0
ES	5	320	0
Europe	1	0	1
FI	16	25282	6
GB	52	74014	5
IE	1	1450	0
IL	1	10	0
IS	1	564	0
IT	1	1469	0
JP	8	117	3
NL	2	455	1
NO	9	11549	3
NZ	1	1510	0
PT	1	30	0
SE	8	18989	3
US	83	289054	19
root	6	872	0
Total	266	501456	59

January, 1992

PSI participated in the NADF meeting held in mid January, with participants of all organizations providing an X.500 based directory service.

Preparation continues for our participation in the NADF Experimental Pilot, to provide an industry-wide set of standards for directory services which will make most primary applications compatible. PSI helped test the software which will be used for the NADF CAN and act as a central clearing house for information. We participated in the January meeting where the NADF Knowledge And Naming (KAN) software was redesigned to better interact with CAN. Both KAN and CAN software must now be modified to provide that compatibility. The change includes provisions to provide updates from DSAs to the NADF CAN, and to accept updates from the CAN to DSAs.

Work continues on software to make some of the information from the White Pages Pilot Project available in the NADF Experimental Pilot.

February, 1992

Work has continued on the White Pages and related X.500 software development projects, conducted by Wengyik Yeong, PSI Scientist, under the direction of Martin Schoffstall, Chief Technical Officer.

Current White Pages status:

- WPP participants are being encouraged to move from the c=US Master to a slave of c=US which we also run. This is an effort to do some load-balancing between the two, the idea being that DSAs in the U.S. will get slave copies of U.S. information from a U.S. slave, freeing the U.S. master to service X.500 queries. [At a very coarse level, there are two kinds of interactions between DSAs: requests to send all information on a certain level of the tree, and requests to find things in the tree. Due to the fact that the c=US master is having serious problems servicing both kinds of requests, WPP participants are being encouraged to send the two types of requests to two different DSAs]

- transition to the NADF naming scheme is still ongoing. Most people have experienced difficulty of one type or another, mostly due to a lack of understanding of the NADF scheme's rules.

- we are continuing to get copies of the information directly below national nodes in the global tree so that requests for foreign information can be serviced locally (in the U.S.). The most recent addition was l=Europe, which we started keeping copies of because nobody in Europe (other than the British) was willing to service requests for it seriously. We currently hold copies of the root of the tree, c=SE, c=PT, c=NZ, c=NO, c=NL, c=JP, c=IT, c=IS, c=IL, c=IE, c=GB, c=FI, c=ES, c=DK, c=DE, c=CH, c=CA, c=BE, c=AU, o=COSINE, l=Europe on both the U.S. master and the U.S. slave that we run.

- Mr. Yeong is currently following a number of problems which are affecting DSA reliability (specifically the reliability of the U.S. master and slave). These include: DSAs going into a coma under extreme load, DSAs spontaneously rebooting themselves for no apparent reason, problems people have been experiencing getting updates of the information under c=US (one solution has been to offload this to the U.S. slave, as above) and the Directory Assistance server on the wp.psi.net service machine dying under heavy load.

Other than the maintenance of the White Pages pilot, which is a significant portion of labor, this project has also supported the following tasks:

- development of software to load DNS information into the X.500 tree
- modifications and bug fixes to the NADF KAN software
- software to produce KAN updates from the X.500 tree [KAN: "Knowledge and Naming"]
- work to specify the Lightweight Directory Browsing Protocol (LDBP) (An LDBP implementation is "future work").
- work to specify attribute syntax encodings for the use of the Lightweight Directory Protocols (of which LDBP above is one)
- the 'usconfig' program to generate White Pages configurations that conform to the NADF naming scheme (past work).
- the Macintosh and DOS 'psiwp' programs (past work).

- ongoing work on improving the quipu X.500 software with bug reports, fixes and testing of beta versions of the software (past, present and future work).
- work reviewing countless drafts of countless documents concerning various aspects of the Internet X.500 pilot and the NADF Directory work (past, present and future work).

```
----------------------------------------------------
Current List of WhitePages Participants:
----------------------------------------------------

Connected to Spectacled Bear at '0101'H/Internet=192.67.6.2+17003
Current position: @c=US@o=Performance Systems International
User name: @
Current sequence: default
DAP-listener: 127.0.0.1 36046
1 o=Advanced Decision Systems
2 o=Anterior Technology
3 o=Apple Computer, Inc.
4 o=Auburn University
5 o=Bellcore
6 o=Bucknell University
7 o=Carnegie Mellon University
8 o=Case Western Reserve University
9 o=Clarkson University
10 o=Columbia University
11 o=Contel Federal Systems
12 o=Control Data
13 o=Corporation for National Research Initiatives
14 o=Cray Research Inc.
15 o=Dana Farber Cancer Institute
16 o=Defense Communications Agency
17 o=DMD
18 o=Dover Beach Consulting, Inc.%st=California
19 o=Duke University
20 o=Eastman Kodak Co.
21 o=Energy Sciences Network
22 o=Florida State University
23 o=GTE Laboratories, Inc.
24 o=Hewlett-Packard
25 o=Intel Corp.
26 o=IntelliGenetics, Inc.
27 o=INTERACTIVE Systems
28 o=Lawrence Berkeley Laboratory
29 o=Lawrence Livermore National Laboratory
30 o=MCNC
31 o=Merit Computer Network
32 o=Michigan State University
```

33 o=National Aeronautics and Space Administration
34 o=National Energy Research Supercomputer Center
35 o=National Institutes of Health
36 o=National Science Foundation
37 o=NCI Supercomputer Center
38 o=Network Management Associates%st=California
39 o=New Mexico State University
40 o=New York University
41 o=Northern Telecom
42 o=NorthWest Net
43 o=Oakland University
44 o=Performance Systems International
45 o=Portland State University
46 o=Princeton University
47 o=Rensselaer Polytechnic Institute
48 o=Rockefeller University
49 o=Schlurnberger
50 o=Sprintmail
51 o=SRI International
52 o=State University of New York at Buffalo
53 o=State University of New York at Stony Brook
54 o=Sun Microsystems Incorporated
55 o=Syracuse University
56 o=Texas A+M University
57 o=The MITRE Corporation
58 o=TRW Inc.
59 o=U.S. Government
60 o=United States Army
61 o=University of Alaska Fairbanks
62 o=University of Colorado at Boulder
63 o=University of Colorado at Denver
64 o=University of Florida
65 o=University of Michigan
66 o=University of Minnesota
67 o=University of Pittsburgh
68 o=University of Rochester
69 o=University of Tennessee
70 o=University of Wisconsin
71 o=Virginia Tech
72 o=Vitalink Communications Corporation
73 o=Yale University
74 o=Department of Air Force
75 o=Department of Energy
76 o=Department of Air Force@o=Rome Laboratory
77 o=Dover Beach Consulting, Inc.
78 o=Information Sciences Institute

79 o=Network Management Associates
80 o=Stanford Linear Accelerator Center
81 o=Stanford University
82 o=Mississippi State University
83 o=University of Mississippi
84 o=Washington University
85 o=University of Nebraska-Lincoln
86 o=Rutgers University
87 o=Alfred University
88 o=City College of CUNY
89 o=Cornell University
90 o=NYSERNet Inc.
91 o=Polytechnic University
92 o=Rochester Institute of Technology
93 o=State University of New York at Albany
94 o=University of Oregon
95 o=University of Pennsylvania
96 o=University of Houston

March, 1992

Work has continued on the White Pages and related X.500 software development projects, conducted by Wengyik Yeong, PSI Scientist, and Chris Kolb, Programmer, under the direction of Martin Schoffstall.

In response to bug reports from early testers of the software used to load DNS zone files into the DIT, changes were made to the software.

PSI participated in Directory Services activities at IETF in San Diego.

The software to generate NADF KAN updates was completed. Comments about the KAN update procedure and the update format were provided to the NADF based on implementation experience.

PSI requested and received responses to comments about the initial draft of the Lightweight Directory Browsing Protocol, mentioned in previous reports. The comments received at the IETF meeting are currently being considered while the document undergoes revision.

A March Supplemental Report was submitted with a presentation: Comments on NADF Agreements for Name and Knowledge Sharing, by Wengyik Yeong, included as Appendix F.

April, 1992

PSI's scientist, Wengyik Yeong, continued work on X.500 in support of this project. Much of the work performed this month was related to following up on the Internet Engineering Task Force meeting last month. In particular, there were activities which provide a broader base of X.500 users by extending X.500 access to X.400 electronic mail users. Also, design of smaller (lightweight) software has been continuing which will result in a much broader application of X.500 directory services.

The activity this month included:

- A preliminary draft of a MIB that currently instruments X.400 MTAs and X.500 DSAs was reviewed. Comments have been submitted to the author.
- Based on comments received on the LDBP at the IETF meeting, the scope of the protocol was expanded to include simple Directory management, and a new specification, renamed the Lightweight Directory Access Protocol (LDAP) has been resubmitted.
- A new draft of RFC1279[4] containing changes suggested at the IETF meeting was reviewed. A minor modification was also made to the 'fred' software due to a bug that was uncovered during the IETF discussion on the DNS in X.500.

Beta testing was performed on an interim release of the ISODE QUIPU software. Further testing will be performed over the next few months.

PSI participated in the 9th meeting of the NADF.

PSI has co-founded the ISODE Consortium, which will take over management of future ISODE development and distribution. PSI will cooperate fully in this process. Additional information on the Consortium will be included in the next report.

The following organizations joined the PSI White Pages Pilot this month:

Martin Marietta
Indiana University
Harvey Mudd College
University of Akron

May, 1992

In late April, NYSERNet issued a stop-work order effective May 1, 1992 on this software effort while DARPA considered the various options available to it under this agreement. A meeting was convened at DARPA headquarters with officials of Rome Lab, NYSERNet, DARPA, PSI and DISA present.

The meeting covered a review of historical work, current status, and projected work for the remainder of the contract period. Technical, management and financial questions and answers were discussed.

PSI is prepared to re-start work on this effort on or before June 1, if DARPA and Rome Lab wish this to occur. We will resume work when notified by NYSERNet, Inc. After June 1, PSI must consider reassigning the research staff currently assigned to this project to other long term projects. Once this commitment is made, PSI may not easily restart this effort. If the reassignment is completed, PSI will arrange for a smooth contract termination of the software effort.

June, 1992
PSI re-initiated work in mid June, after receiving authorization to begin work on June 1. PSI continued work on an implementation of the Lightweight Directory Access Protocol as previously designed earlier this year, and described in prior reports. Normal White Pages pilot maintenance continued.

July, 1992
The effort under this contract is beginning to gear up again after the shut-down in May. We are currently at about 1/3 of our previous work effort and will remain so until scientific/programmer staff are released from other tasks next month.

The bulk of the effort this month was dedicated to resuscitating the U.S. Master, @cn=Alpaca after the computer host that DSA was running on suffered a severe hardware failure. In bringing the DSA back online, we took the opportunity to upgrade the DSA from version 7.0 to version 8.0 of the quipu software.

Implementation of the Lightweight Directory Access Protocol (LDAP) continues. The draft specification received minimal comment at the latest IETF meeting in Boston. There were, however, a few changes suggested, including a proposal for the LDAP to support strong authentication through the use of Kerberos-based security services. A new draft of the LDAP specification will be circulated to the authors next week incorporating some of the changes proposed.

The following organization joined the White Pages Pilot Project this month:

Lewis and Clark College

August, 1992
The effort under this contract is in maintenance mode while discussions continue with Rome Lab and DARPA on redirection and refocussing within the work scope. We are currently at about 1/4 of our previous effort.

The bulk of the work this month was dedicated to maintaining the U.S. Master, providing general support through email, checking use logs, talking to new organizations considering entering the database, and remaining in contact with other developers in the ISODE/X.500 and SNMP.

We expect the redirection discussions to be completed next month and work to continue at its previous pace.

September, 1992
The effort under this contract remains in maintenance mode while discussions continue with Rome Lab and DARPA on redirection and refocussing within the work scope. We are currently at about 1/4 of our previous effort.

The work this month was dedicated to maintaining the U.S. Master, adding AT&T, Boston College, and the University of Missouri-Rolla, providing general support by helping people set up their DSAs etc., spoke with new organizations considering entering the database, and remaining in contact with other developers in the ISODE/X.500 and SNMP.

In addition, we fixed a configuration file problem on wp.psi.net which supports the White Pages Project. We expect the redirection discussions to be completed in October and work to continue at its previous pace.

October, 1992
The effort under this contract remains in maintenance mode while discussions continue with Rome Lab and DARPA on redirection and refocussing within the work scope. We are currently at about 1/4 of our previous effort.

The work this month was dedicated to maintaining the U.S. Master by making appropriate changes after Merit, Inc. changed DSAs, and providing assistance to participating organizations which had problems with a software bug that was overflowing their disks.

We added three new organizations to the project, including Boston College and ALCOA. We expect the redirection discussions to be completed soon and work to continue at its previous pace in early November.

November, 1992
The effort under this contract remains in maintenance mode while discussions continue with Rome Lab and DARPA on redirection and refocussing within the work scope.

The work this month was dedicated-to maintaining the U.S. Master, consolidating two existing DSAs and moving them to osi.nyser.net, a machine with more resources. The common user interface, "fred", on wp.psi.net was reconfigured to use a local DSA on wp.psi.net, to improve performance.

The following organizations joined the pilot this month:

The University of Texas at Austin
The University of Texas System
Fujitsu Network Switching
COSMIC

December, 1992

Based on Rome Lab direction, NYSERNet instructed PSI to complete the R&D software effort as of 31 December 1992. ARPA and Rome Lab feel that this effort has met its goals and has resulted in numerous software contributions in underlying technology and application within the SNMP, ISODE and X.500 Directory areas. Work has commenced on documenting the overall software effort.

Maintenance of the White Pages Pilot project will continue through the end of January, 1993. At that time, PSI will determine whether it will continue some portions of the WPP on internal funds or discontinue the effort.

Work during December was dedicated to maintaining the U.S. Master DSA and related support work.

Section 4
Conclusions and Recommendations

This section highlights the major conclusions and recommendation resulting from the work performed under this contract.

4.1 Conclusions

- POP-based switching and routing is a successful design for the development of a nationwide high speed computer data network. The design is reliable, flexible, and economical. Such a design depends on access to the POP and freedom of action in the POP. The POP approach will be adopted by all wide area networking systems to the maximum extent allowed by the telephone company carriers.

- The telephone company providers are, today, still learning about software applications being used on wide area networking systems. Therefore, systems integrators and network engineers must be employed to make maximum use of the standard telephone company products to advance research and development.

- Networking groups in federal agencies - such as DOE, NASA, NSF, and DARPA have different missions. An objective of each, therefore, is to direct its own activities. An organization that undertakes to serve many such agencies must be aware of the agencies' different requirements and need for control. Specific issues that must be addressed for each agency include scheduling, technical approaches, and priorities. The requirement that the service accommodate different demands can reduce the inherent benefits of the network's economy of scale. The network's costs, therefore, will be higher if it accommodates the different agencies' requirements than if the same service were provided to each.

4.2 Recommendations

- The significant advantages of reliability and economy of POP-based switching in production-level, in contrast to testbed, networking systems should be provided to all federal agencies for their daily operational activities that require data networking.

- Additional research should be funded to examine the efficacy of new network technologies for widely distributed communities of users, for example SONET and ATM, such as were involved in and demonstrated by this contract.

- Collaborative networking operations should be developed only among agencies that cooperate and fully fund the effort. Their pledge to the effort should be recorded in a written contract. The experience of this contract demonstrates that agencies do not wish to participate in an experiment of shared infrastructure if they believe they might lose any control of their networking operation.

Section 5
References

1. RFC1202, "A PC front-end for the Directory Assistance Service" (Available from the NSF InterNIC at ds.internic.net in the rfc directory)

2. Yeong, W., PSI, Inc., "PSI White Pages Project: Administrator's Guide", For copies, send email to yeongw@psi.com.

3. Yeong, W., PSI, Inc., "Representing Public Archives in X.500", 23 Oct 91, For copies, send email to yeongw@psi.com.

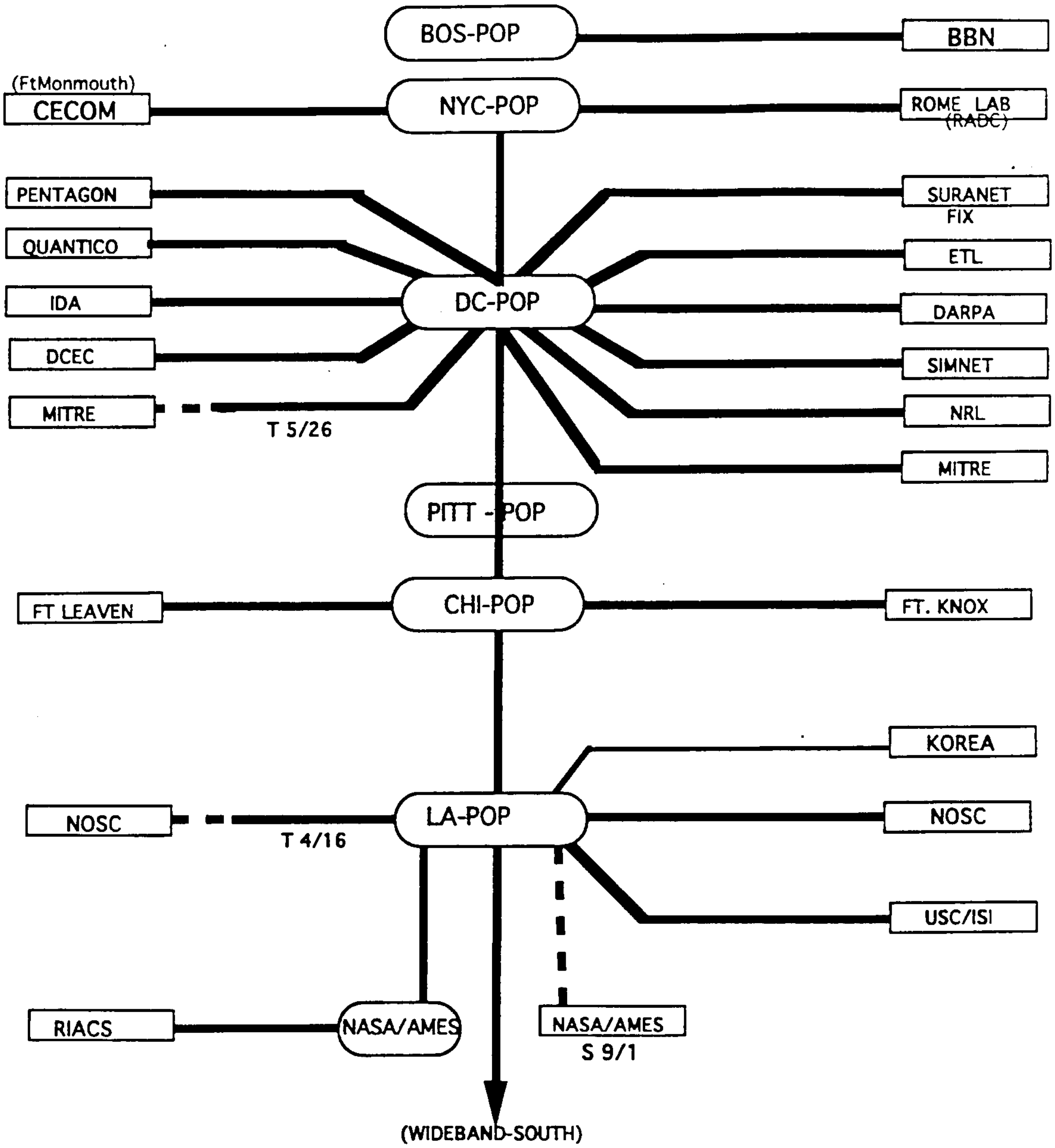
4. RFC1279, "X.500 and Domains", Hardcastle-Kille, S.E. (Available from the NSF InterNIC at ds.internic.net in the rfc directory)

Appendix A

NNT Topology Supporting Six Networks (TWBnet, CORNETT, ESNET, DARTNet, Los Nettos, and Arpanet)

28 May 1992

WIDEBAND (TWBnet)



NATIONAL NETWORK TESTBED (NNT)

PSI/NYSERNET

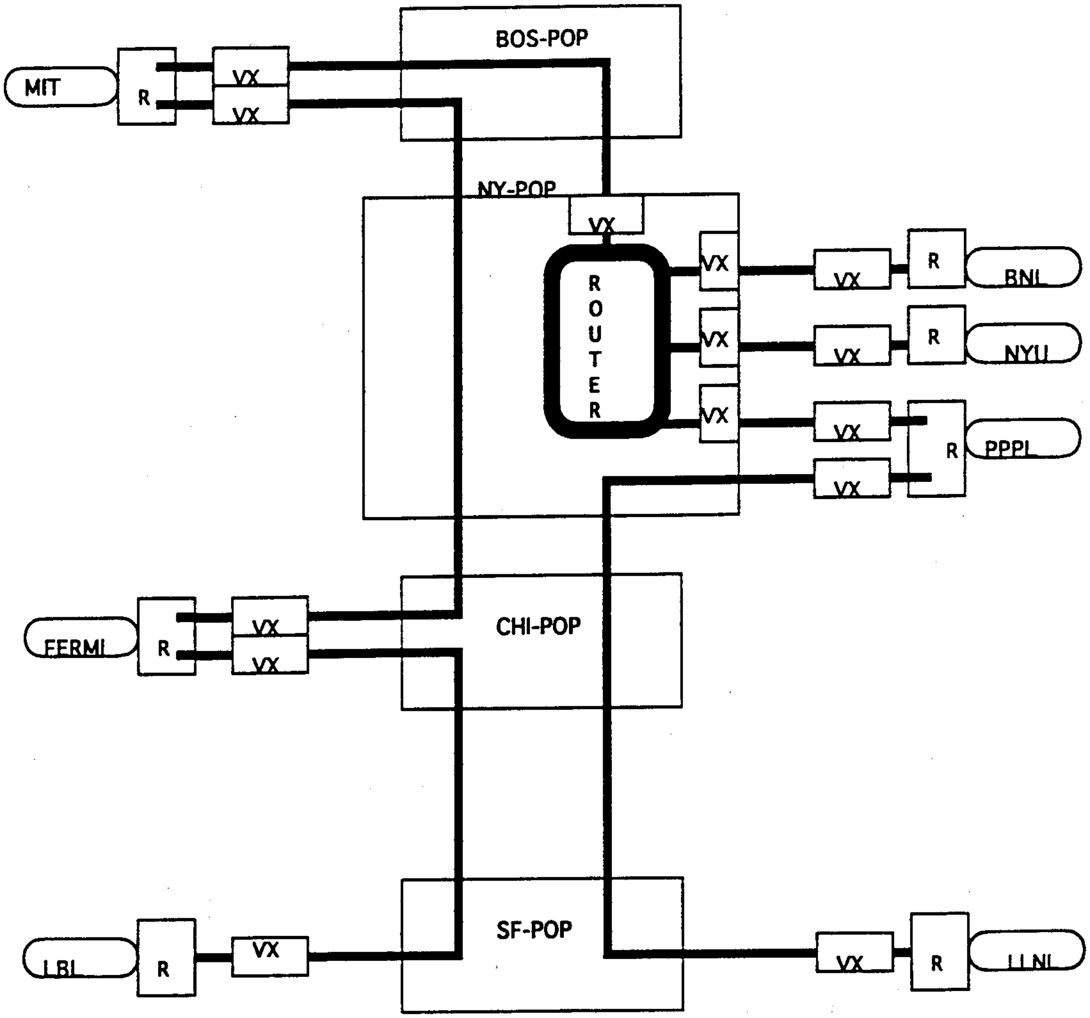
▬ ▬ ▬ IN PROCESS

⬭ CPE Sites

ESnet (DOE)

27 JUNE 1991

NOTE: ALL circuits scheduled for DISCONNECT on 1 JULY 1991.



NATIONAL NETWORK TESTBED (NNT)

NYSERNET

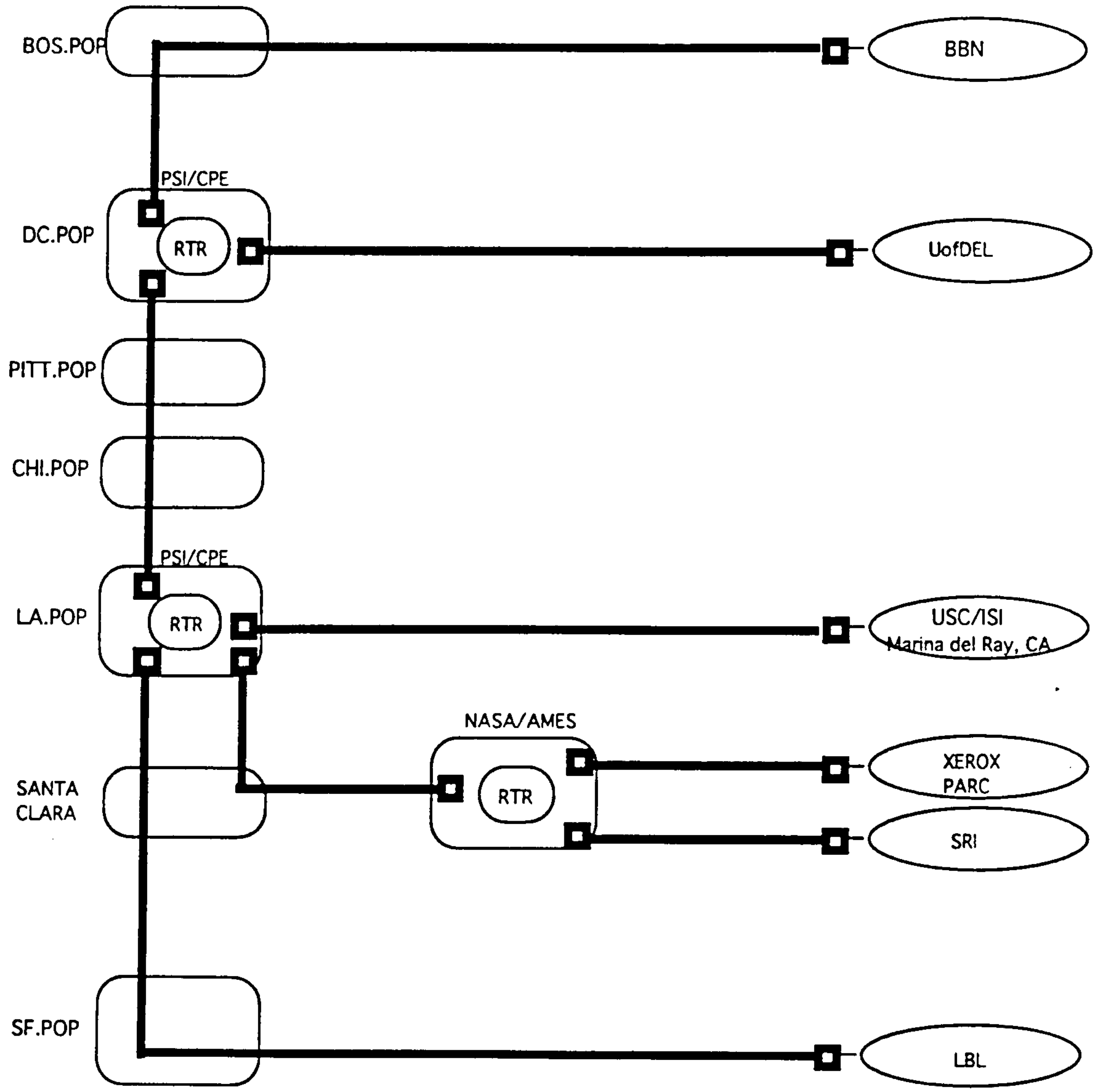
Legends:
VX=CSU/DSU
R =Router

DARTNet

(formerly DRI-2)

29 SEP 1992

DARTnet is scheduled for disconnect - 19 October 1992



NATIONAL NETWORK TESTBED (NNT)

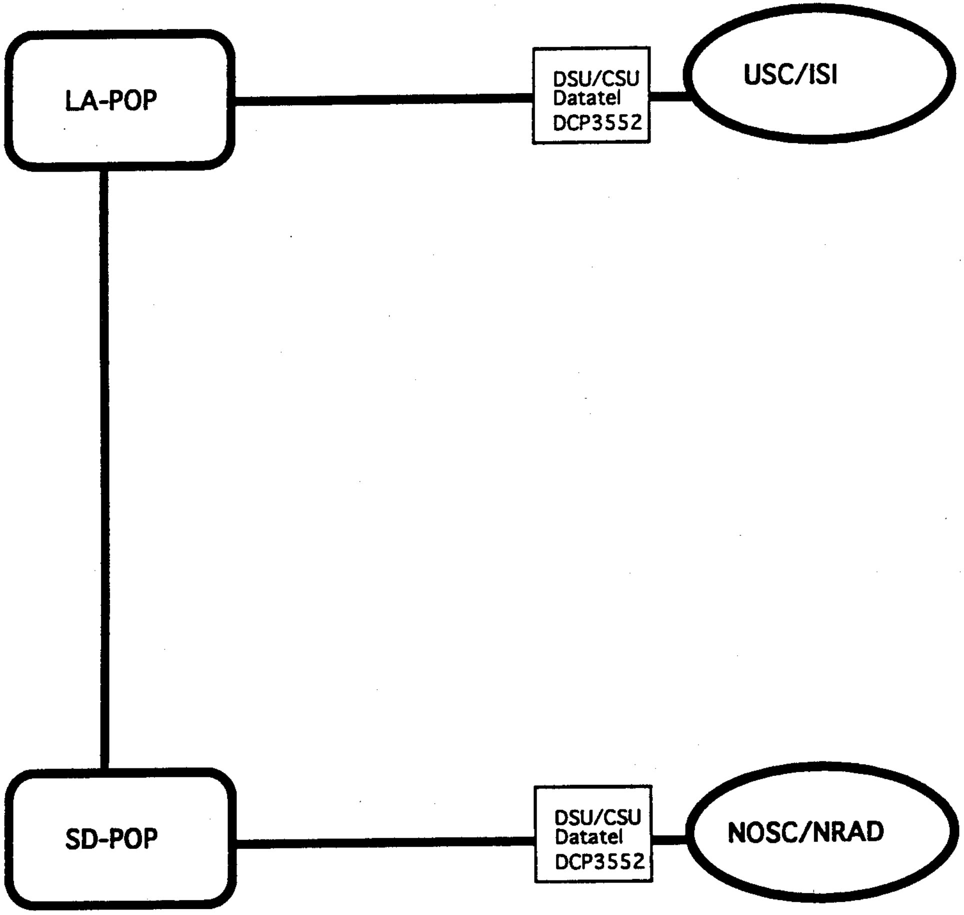
PSI/NYSERNET

Legend: ▣ = DL551VX (RTR) = DARTnet Router

LOSNETTOS

28 Jan 1993

Disconnect requested 26 Jan 93



LA-POP

DSU/CSU Datatel DCP3552

USC/ISI

SD-POP

DSU/CSU Datatel DCP3552

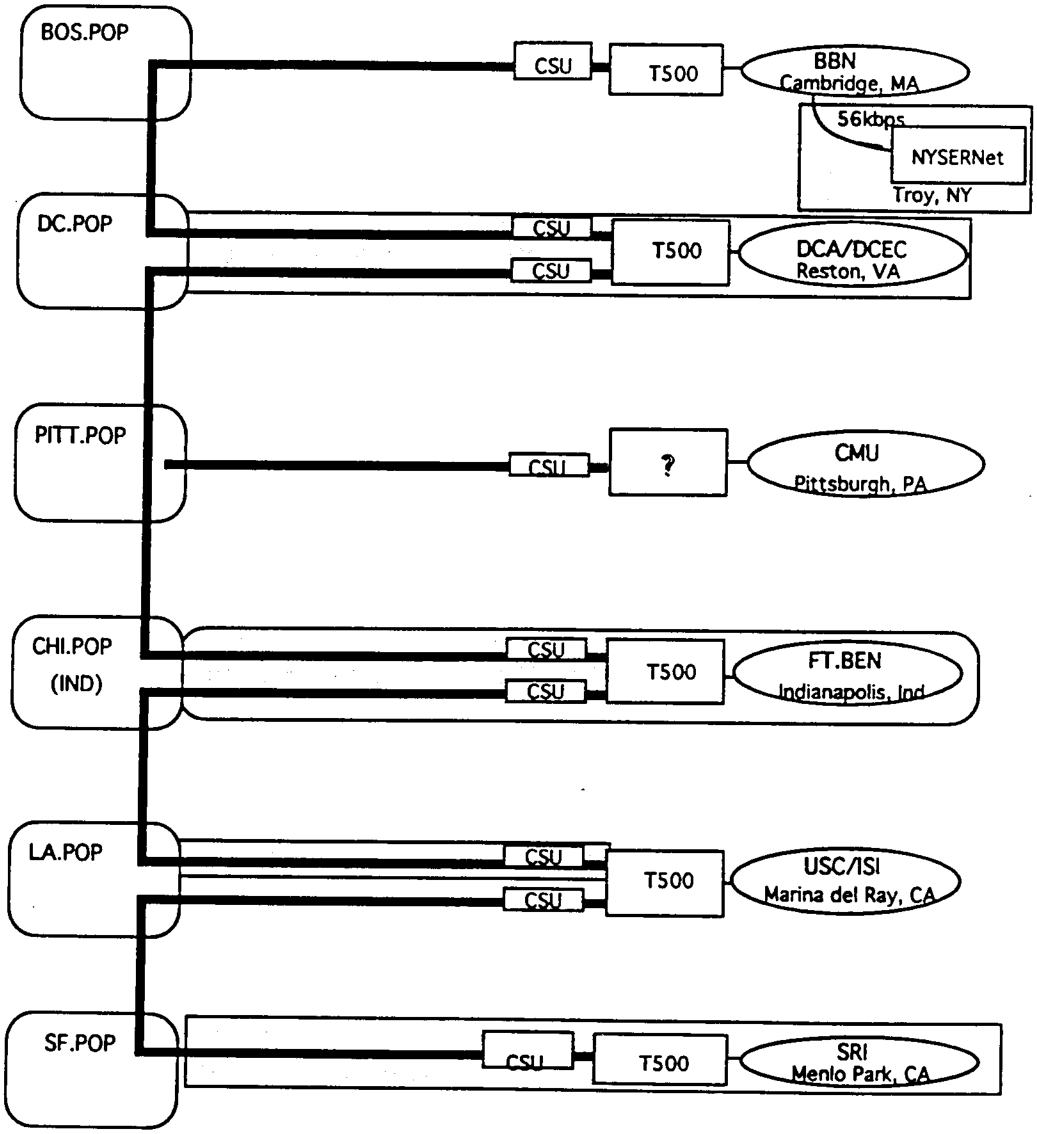
NOSC/NRAD

NATIONAL NETWORK TESTBED (NNT)

NYSERNET

ARPANET

As of: 21 Mar 1990



NATIONAL NETWORK TESTBED (NNT)

NYSERNET

Legend: □ = Cncellation in process.

Appendix B

Current Participants in the White Pages Project

Current Participants in the White Pages Project

Advanced Decision Systems, US
Anterior Technology, US
Apple Computer, Inc., US
ATT, US
Auburn University, US
Bellcore, US
Bucknell University, US
Carnigie Mellon University, US
Case Western Reserve University, US
Citibank, US
Clarkson University, US
Columbia University, US
Continuous Electron Beam Accelerator Facility, US
Control Data, US
Corporation for National Research Initiatives, US
Cray Research Inc., US
CREN, District of Columbia, US
Dana Farber Cancer Institute, US
Defense Communications Agency, US
DMD, US
Dover Beach Consulting, Inc., California, US
Duke University, US
Eastman Kodak Co., US
Energy Sciences Network , US
Florida State University, US
GTE Laboratories, Inc., US
Hewlett-Packard, US
IntelliGenetics, Inc., US
INTERACTIVE Systems, US
ISODE Consortium, GB
Lawrence Berkeley Laboratory, US
Lawrence Livermore National Laboratory, US
Martin Marietta + Maryland, US
MCNC, US
Merit Computer Network, US
Michigan State University, Michigan, US
National Aeronautics and Space Administration, US
National Center for Manufacturing Sciences + Michigan, US
National Energy Research Supercomputer Center, US
National Institutes of Health, US
National Library of Medicine, US
National Science Foundation, US
NCI Supercomputer Center, US
Network Management Associates, California, US

New Mexico State University, US
New York University, US
Northern Telecom, US
NorthWest Net, US
Oakland University, US
Pacific Northwest Laboratory, US
Performance Systems International, US
Portland State University, US
Princeton University, US
Rensselaer Polytechnic Institute, US
Rockefeller University, US
Rockwell, US
Sandia National Laboratories, US
Schlumberger, US
Sprintmail, US
SRI International, US
State University of New York at Albany, New York, US
State University of New York at Buffalo, US
State University of New York at Stony Brook, US
Sun Microsystems Incorporated, US
Texas A +M University, US
The MITRE Corporation, US
The University of Texas System, US
TRW Inc., US
DMD, US
U.S. Government, US
United States Army, US
University of Alaska Fairbanks, US
University of Colorado at Boulder, US
University of Colorado at Denver, US
University of Florida, US
University of Michigan, US
University of Minnesota, US
University of Pittsburgh, US
University of Rochester, US
University of Wisconsin, US
Virginia Tech, US
Yale University, US
Department of Air Force, U.S. Government, US
Department of Energy, U.S. Government, US
Department of Navy, U.S. Government, US
Rome Laboratory, Department of Air Force, U.S. Government, US
Naval Research Laboratory, Department of Navy, U.S. Government, US
Arizona State University, Arizona, US
Dover Beach Consulting, Inc., California, US
Harvey Mudd College, California, US

Information Sciences Institute, California, US
Intel Corporation, California, US
INTEROP, California, US
Network Management Associates, California, US
Stanford Linear Accelerator Center, California, US
Stanford University, California, US
University of California at Los Angeles, California, US
Vitalink Communications Corporation, California, US
University of Colorado at Bolder, Colorado, US
OSODE Consortium, GB
CREN, District of Columbia, US
ISODE Consortium, GB
COSMIC, Georgia, US
Indiana University, Indiana, US
Boston College, Massachusetts, US
Digital Equipment Corporation, Massachusetts, US
Massachusetts Institute of Technology, Massachusetts, US
Michigan State University, Michigan, US
Mississippi Center for Supercomputing Research, Mississippi, US
Mississippi State University, Mississippi, US
University of Missouri - Rolla, Missouri, US
Washington University, Missouri, US
University of Nebraska-Lincoln, Nebraska, US
Rutgers University, New Jersey, US
Alfred University, New York, US
City College of CUNY, New York, US
NYSERNet, Inc, New York, US
Polytechnic University, new York, US
Rochester Institute of Technology, New York, US
State University of New York at Albany, New York, US
Syracuse University, New York, US
Fujitsu Network Switching, North Carolina, US
The University of Akron, Ohio, US
Lewis and Clark College, Oregon, US
University of Oregon, Oregon, US
Alcoa Technical Center, Pennsylvania, US
Pennsylvania State University, Pennsylvania, US
University of Pennsylvania, Pennsylvania, US
ISODE Consortium, GB
Microelectronics and Computer Technology Corporation, Texas, US
The University of Texas at Austin, Texas, US
University of Houston, Texas, US
McDonnell Douglas Corporation, Virginia, US
SofTech, Inc., Virginia, US

Appendix C

Application Monitoring MIB

Network Working
Group
INTERNET-DRAFT

S.E. Hardcastle-Kille
ISODE Consortium
T. Lenggenhager
SWITCH
D. Partain
University of Linkoping
W. Yeong
Performance

October 17, 1992
Expires: April 1993

Application Monitoring MIB

Status of this Memo

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a "working draft"or "work in progress." Please check the I-D abstract listing contained in each Internet Draft directory to learn the current status of this or any other Internet Draft.

Abstract

This document defines a MIB for monitoring applications running on a system. It defines specific attributes for MTAs and DSAs, and this approach could be easily extended to other applications.This document is an agreed ISODE Consortium specification (IC 4 (Version 3.12)).

1 Why?

There is a substantial need to monitor applications, particularly distributed system components such as MTAs and DSAs, in order to determine heavy load, broken connectivity, system failure or congestion. Specific requirements are:

- General monitoring of a large number of components (typical for a large organisation).
- Integration with general network management.

SNMP is the clear choice for this function. The main goal is very simple read-only access. Essentially to determine up/down status and indicate operational problems (typically heavy load).

1.1 Restricted Scope

There is a lot more that could be done. For example:

- General MTA reconfiguration
- Examination and modification of mail queues

- Probing to find location of messages which have left the local MTA but have not been delivered

Whilst this is cute, to be effective, it requires security. It will also be a lot more contentious and have awkward choices between generic and implementation-specific aspects. There are also other reasonable approaches to this sort of problem. This document will religiously keep simple and focus on the basic monitoring aspect.

2 Relationship to Directory

Use of (X.500) directory already is tied up with application management. There are clearly many things which could be dealt with by directory or management protocols. We take the line here that static configuration is dealt with in the directory, and dynamic by

management protocols.
By placing the static information in the directory, the richness and linkage of the directory information framework does not need to be repeated in the MIB. Static information is information which has a mean time to change of the order of days or longer.
A linkage will be established, so that:

- The managed object contains its own directory name. This allows all directory information to be obtained by reference. This will allow a Directory capable SNMP monitor to present this information to the manager. It is intended that this will be the normal case.

- The directory will reference the location of the SNMP agent, so that an SNMP capable DUA could probe dynamic characteristics of the object.

- This approach could be extended further, so that the SNMP attributes are modelled as directory attributes. This would allow an SNMP capable DSA to present this information to a standard DUA.

3 Application Objects

This MIB starts with a set of general purpose attributes which would be appropriate for a range of network applications, both OSI and non-OSI. Subsequent sections give attributes specific to applications. A table is defined which will have one row for each application running on the system. The only static information held on the application is its distinguished name. All other static information can be determined from the directory. The Directory Name is an external key, which allows an SNMP MIB entry to be cleanly related to the X.500 Directory. In SNMP terms, the applications are grouped in table (applicationTable), which is indexed by an integer key (applicationIndex).

The type of the application will be determined by one or both of:

- Additional MIB variables specific to the applications

- An association to the application of a specific protocol

applicationTable OBJECT-TYPE

```
  SYNTAX SEQUENCE OF ApplicationEntry
  ACCESS not-accessible
  STATUS mandatory
  DESCRIPTION
"The table holding objects which apply to all different kinds
of applications. At present, it holds information for DSA's
and MTA's"
::= {application-mib 1}


applicationEntry OBJECT-TYPE
SYNTAX ApplicationEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "Entry associated with application"
  INDEX { applicationIndex }
  ::= {applicationTable 1}

ApplicationEntry ::= SEQUENCE {
applicationIndex
     INTEGER,
distinguishedName
    -DisplayString,
applicationStatus
     INTEGER,
applicationUptime
    TimeTicks,
inboundAssociations
     INTEGER,
outboundAssociations
    INTEGER,
accumulatedInboundAssociations
   Counter,
accumulatedOutboundAssociations
   Counter,
lastInboundActivity
   TimeTicks,
lastOutboundActivity
   TimeTicks,
failedOutboundAssociations
    Counter
}
```

```
applicationIndex OBJECT-TYPE
   SYNTAX INTEGER
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
     "Index to uniquely identify the application"
   ::= {applicationEntry 1}


distinguishedName OBJECT-TYPE
SYNTAX DisplayString
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The string encoded distinguished name of the managed object
using the format of OSI-DS 23"
::= {applicationEntry 2}


applicationStatus OBJECT-TYPE
SYNTAX INTEGER {
down(1),
running(2),
halted(3),
congested(4),
restarting(5)
  }

 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
  "Indicates the operational status of the application entity"
 ::= {applicationEntry 3}

applicationUptime OBJECT-TYPE
 SYNTAX TimeTicks
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
  "The time  since the application was initialised"
  ::= {applicationEntry 4}

inboundAssociations OBJECT-TYPE
   SYNTAX INTEGER
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
      "The number of current associations to the application entity, where
      it is the responder. For dynamic single threaded processes,
      this will be the number of application instances"
   ::= {applicationEntry 5}


outboundAssociations OBJECT-TYPE
   SYNTAX INTEGER
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
      "The number of current associations to the application entity, where
       it is the initiator.  For dynamic single threaded processes,
       this will be the number of application instances"
   ::= {applicationEntry 6}


accumulatedInboundAssociations OBJECT-TYPE
   SYNTAX Counter
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
      "The total number of associations to the application entity since
       application initialisation, where it is the responder. For
       dynamic single threaded processes, this will be the number
       of application instances"
   ::= {applicationEntry 7}


accumulatedOutboundAssociations OBJECT-TYPE
   SYNTAX Counter
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
      "The total number of associations to the application entity since
       application initialisation, where it is the initiator. For
       dynamic single threaded processes, this will be the number
       of application instances"
   ::= {applicationEntry 8}
```

```
lastInboundActivity OBJECT-TYPE
   SYNTAX TimeTicks
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
    "The time since this application has had an inbound association."
   ::= {applicationEntry 9}

lastOutboundActivity OBJECT-TYPE
   SYNTAX TimeTicks
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
    "The time since this application has had an outbound association."
   ::= {applicationEntry 10}

failedOutboundAssociations OBJECT-TYPE
   SYNTAX Counter
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
      "The total number associations where the application entity is
        initiator and association establishment has failed, since
        application initialisation"
   ::= { applicationEntry 11}
```

The basic applicationTable contains a list of the application entities. A second table is maintained, which holds the list of associations. This is treated as a separate group to the basic application table. Where simplified appplication monitoring is needed, this group may be omitted. This table is indexed by applicationIndex and associationIndex, with the application index coming first.

```
associationTable OBJECT-TYPE
   SYNTAX SEQUENCE OF AssociationEntry
   ACCESS not-accessible
   STATUS mandatory
   DESCRIPTION
     "The table holding a set of all active application associations"
   ::= {application-mib 4}
```

```
-- the table is indexed through a combination of the index into the
-- application table and an index unique to this table. for instance,
-- if you were to "get remoteApplication.0.0", it would be the instance
-- associated with the first entry in the applicationTable and the first
-- entry in this table for that application.


associationEntry OBJECT-TYPE
   SYNTAX AssociationEntry
   ACCESS not-accessible
   STATUS mandatory
   DESCRIPTION
    "Entry associated with association"
   INDEX { associationApplicationIndex, associationIndex}
   ::= {associationTable 1}

AssociationEntry ::= SEQUENCE {
   associationApplicationIndex
      INTEGER,
   associationIndex
      INTEGER,
   remoteApplication
      DisplayString,
   applicationProtocol
      OBJECT IDENTIFIER,
   applicationType
      INTEGER,
   associationDuration
      TimeTicks
}

associationApplicationIndex OBJECT-TYPE
   SYNTAX INTEGER
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
      "Index into the application table to identify the (local)
      application associated with the association and also an
      index on this table"
   ::= {associationEntry 1}

associationIndex OBJECT-TYPE
```

```
   SYNTAX INTEGER
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
    "Index to uniquely identify the association"
   ::= {associationEntry 2}


remoteApplication OBJECT-TYPE
   SYNTAX DisplayString
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
       "The name of the remote application. For an internet application
       this should be a domain name. For an OSI application it should be
       the string encoded distinguished name of the managed object
       using the format of OSI-DS 23.
       For X.400(84) MTAs which do not have a Distinguished Name, the
       RFC 1327 syntax 'mta in globalid' should be used"
   ::= {associationEntry 3}


applicationProtocol OBJECT-TYPE
   SYNTAX OBJECT IDENTIFIER
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
       "An identification of the protocol being used for the
       application. For an OSI Application, this will be the
       Application Context. For Internet applications, it should be an
       Object Identifier derived from the port ** see below ***"
   ::= {associationEntry 4}


applicationType OBJECT-TYPE
   SYNTAX INTEGER {
       ua-initiator(1),
       ua-responder(2),
       peer-initiator(3),
       peer-responder(4) }
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
       "Shows whether the remote application is a User Agent, or a peer
```

```
      server, and whether the remote end is initiator or responder"
  ::= {associationEntry 5 }


associationDuration OBJECT-TYPE
   SYNTAX TImeTicks
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
      "The time that the association has been open"
   ::= {associationEntry 6}
```

Editor's Note: It has been suggested that a table be kept of failed associations, in order to help detect remote applications which are unavailable. Comments are solicited for future versions of this document.
A count of failed associations is used here. This does not have the potential growth problems of a table. If this count is excessive, the logs may be examined.

4 MTA Objects

If there is an MTA on the host, the following mta group may be used to monitor it. Only one MTA may be monitored on a host. This restriction is made in order to simplify the MIB. In the rare case of running multiple MTAs on one host, they may both be monitored by monitoring one from a different host which has no MTA.
The first parameters are per-MTA parameters.

```
mta OBJECT IDENTIFIER ::= {application-mib 2}

mtaApplicationIndex OBJECT-TYPE
   SYNTAX INTEGER
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
      "Reference into application table to allow correlation with
        general application parameters"
  ::= {mta 1}
```

```
numberMessages OBJECT-TYPE
   SYNTAX INTEGER
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
     "The total number of messages in all the MTA queues"
 ::= { mta 2}

volumeMessages OBJECT-TYPE
   SYNTAX INTEGER
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
      "The total volume of messages in all the MTA queues, measured
       in kbytes"
   ::= {mta 3}

submittedMessages OBJECT-TYPE
   SYNTAX Counter
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
      "The number of messages submitted since application
      initialisation"
   ::= {mta 4}


deliveredMessages OBJECT-TYPE
   SYNTAX Counter
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
      "The number of messages delivered since application
       initialisation"
   ::= {mta 5}

lastInboundMtaActivity OBJECT-TYPE
   SYNTAX TimeTicks
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
      "Time since the last time that this MTA had an active
      inbound association from a remote MTA"
```

```
  : := {mta 6}

lastOutboundMtaActivity OBJECT-TYPE
  SYNTAX TimeTicks
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
     "Time since the last time that this MTA had an
     outbound association to a remote MTA"
  ::= {mta 7}
```

In addition to representing the MTA, and per-MTA information, there is a table which holds information on every remote MTA for which the monitored MTA has messages queued.

```
queuedMtaTable OBJECT-TYPE
  SYNTAX SEQUENCE OF QueuedMtaEntry
  ACCESS not-accessible
  STATUS mandatory
  DESCRIPTION
     "The table holding information specific to each queue for a
     given remote MTA"
  ::= {application-mib 5}


queuedMtaEntry OBJECT-TYPE
  SYNTAX QueuedMtaEntry
  ACCESS not-accessible
  STATUS mandatory
  DESCRIPTION
   "Entry associated with each remote MTA"
  INDEX { mtaIndex }
  ::= {queuedMtaTable 1}

QueuedMtaEntry ::= SEQUENCE {
  mtaIndex
     INTEGER,
  queuedMtaAssociationIndex
     INTEGER,
  numberMessagesForMTA
     INTEGER,
  volumeMessaqesForMTA
```

```
        INTEGER,
    oldestMessageQueued
        TimeTicks,
    connectFailureReason
        DisplayString,
    lastInboundRemoteMtaActivity
        TimeTicks,
    lastOutboundRemoteMtaActivity
        TimeTicks,
    scheduledRetry
        TimeTicks,
    remoteMTA
        DisplayString,
    mailProtocol
        OBJECT IDENTIFIER
}

mtaIndex OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
    "Index of queued MTAs"
    ::= {queuedMtaEntry 1}

queuedMtaAssociationIndex OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Reference into association table to allow correlation with
        active association(s) on queue. If there is no active
        association for this MTA, it should be -1"
    ::= {queuedMtaEntry 2}


numberMessagesForMTA OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
      "The number of messages queued for the remote MTA"
    ::= {queuedMtaEntry 3}
```

```
volumeMessagesForMTA OBJECT-TYPE
   SYNTAX INTEGER
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
     "The volume of messages queued for the remote  MTA, measured in kbytes"
   ::= {queuedMtaEntry 4}


oldestMessageQueued OBJECT-TYPE
   SYNTAX TimeTicks
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
       "The age of the oldest message queued for this MTA, relative
       to the time of query"
   ::= {queuedMtaEntry 5}


connectFailureReason OBJECT-TYPE
   SYNTAX DisplayString
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
       "The failure reason, if any, for the last connect attempt to
       the MTA. An empty string implies that the last connection
       attempt was successful. If there was
       no connection since the application started it should
       contain 'never' "
   ::= {queuedMtaEntry 6}


lastInboundRemoteMtaActivity OBJECT-TYPE
   SYNTAX TimeTicks
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
       "Time since the last time this remote MTA had an inbound
       association to the local MTA"
   ::= {queuedMtaEntry 7}
```

```
lastOutboundRemoteMtaActivity OBJECT-TYPE
   SYNTAX TimeTicks
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
       "Time since the last time this remote MTA had an outbound
       association to the local MTA"
   ::= {queuedMtaEntry 8}


scheduledRetry OBJECT-TYPE
   SYNTAX TimeTicks
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
       "The scheduled time at which the next connection will be
       attempted. This time is relative to the query time"
   ::= {queuedMtaEntry 9}


remoteMTA OBJECT-TYPE
   SYNTAX DisplayString
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
       "The name of the remote MTA. For an internet application
       this should be a domain name. For an OSI application it should be
       the string encoded distinguished name of the managed object
       using the format of OSI-DS 23.
       For X.400(84) MTAs which do not have a Distinguished Name, the
       RFC 1327 syntax 'mta in globalid' should be used"
   ::= {queuedMtaEntry 10}


mailProtocol OBJECT-TYPE
   SYNTAX OBJECT IDENTIFIER
   ACCESS read-only
   STATUS mandatory
```

```
 DESCRIPTION
    "An identification of the protocol that will be used for the
    connection. For an OSI Application, this will be the
    Application Context. For Internet applications, it should be an
    Object Identifier derived from the port ** see below ***"
 : := {queuedMtaEntry 11}
```

5 DSA Objects

If it is desired to monitor DSAs (Directory System Agents) in more detail, a group of objects is provided for this. The set of DSAs is represented as a table, keyed by dsaApplicationIndex, in order to allow multiple DSAs to be run and monitored on a single host. This situation is sufficiently common to justify this increase in complexity. This key is the same as the applicationIndex, so that generic information may be correlated to DSA information.

```
dsaTable OBJECT-TYPE
   SYNTAX SEQUENCE OF DSAEntry
   ACCESS not-accessible
   STATUS mandatory
   DESCRIPTION
      "The table holding information specific to a DSA"
   ::= {application-mib 3}

dsaEntry OBJECT-TYPE
   SYNTAX DSAEntry
   ACCESS not-accessible
   STATUS mandatory
   DESCRIPTION
     "Entry associated with each DSA"
   INDEX { dsaApplicationIndex }
   ::= {dsaTable 1}

DSAEntry ::= SEQUENCE {
dsaApplicationIndex
      INTEGER,
masterEntries
      INTEGER,
     copyEntries
         INTEGER,
     cacheEntries
         INTEGER,
     readOperations
         Counter,
     searchOperations
         Counter,
     modifyOperations
         Counter

 }

 dsaApplicationIndex OBJECT-TYPE
     SYNTAX INTEGER
     ACCESS read-only
     STATUS mandatory
     DESCRIPTION
         "Reference into application table to allow correlation with
         general application parameters"
     ::= {dsaEntry 1}

 masterEntries OBJECT-TYPE
     SYNTAX INTEGER
     ACCESS read-only
     STATUS mandatory
     DESCRIPTION
       "Number of Entries mastered in the DSA"
     ::= {dsaEntry 2}

 copyEntries OBJECT-TYPE
     SYNTAX INTEGER
     ACCESS read-only
     STATUS mandatory
     DESCRIPTION
       "Number of Entries with systematic (slave) copies maintained
       in the DSA"
     ::= { dsaEntry 3}

 cacheEntries OBJECT-TYPE
     SYNTAX INTEGER
     ACCESS read-only
     STATUS mandatory
```

```
  DESCRIPTION
   "Number of Entries cached (non-systematic copies) in the DSA"
  ::= {dsaEntry 4}


readOperations OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
     "The number of read port (read, compare, abandon) operations
     since application initialisation"
  ::= {dsaEntry 5}

searchOperations OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
     "The number of search port-(search, list) operations since
     application initialisation"
  ::= {dsaEntry 6}

modifyOperations OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
     "The number of modify port (add, delete, modify, modifydn)
     operations since application initialisation"
  ::= {dsaEntry 7}
```

A Object Identifier Assignment

```
ApplicationMib {iso(1) org(3) dod(6) internet(1) private(4)
       enterprises(1) isode-consortium (453) definitions (1)
       application-mib(1)}
DEFINITIONS ::= BEGIN

IMPORTS
       enterprises, OBJECT-TYPE, Counter, DisplayString, TimeTicks
                      FROM RFC1151- SMI;


isode-consortium OBJECT IDENTIFIER ::= {enterprises 453}
application-mib OBJECT IDENTIFIER ::= {isode-consortium 2}


-- *** REPEAT EARLIER DEFINITIONS HERE ***

END
```

A means will be defined to allocate an object identifier to each TCP application. This will be done in consultation with the IANA.

Appendix D

The String Representation of Standard Attribute Syntaxes

Network Working Group
INTERNET-DRAFT

Tim Howes
University of Michigan
Steve Hardcastle-Kille
University College London
W engyik Yeong
Performance Systems International
Colin Robbins
X-Tel Services Ltd.

The String Representation of Standard Attribute Syntaxes

1. Status of this Memo

This draft document will be submitted to the RFC Editor as a standards document. Distribution of this memo is unlimited. Please send comments to the authors, or the discussion group <osi-ds@cs.ucl.ac.uk>.

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts).

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a "working draft" or "work in progress."

Please check the I-D abstract listing contained in each Internet Draft directory to learn the current status of this or any other Internet Draft.

2. Abstract

The lightweight directory protocols require that the contents of AttributeValue fields in protocol elements be octet strings. This document defines the requirements that must be satisfied by encoding rules used to render Directory attribute syntaxes into a form suitable for use in the lightweight directory protocols, then goes on to define the encoding rules for the standard set of attribute syntaxes defined in [1,2] and [3].

The attribute syntax encodings defined in this document are adapted from those used in the QUIPU X.500 implementation. The contributions of the authors of the QUIPU implementation in the specification of the QUIPU

syntaxes [4] are gratefully acknowledged.

3. Attribute Syntax Encoding Requirements.

This section defines general requirements for lightweight directory protocol attribute syntax encodings. All documents defining attribute syntax encodings for use by the lightweight directory protocols are expected to conform to these requirements.

The encoding rules defined for a given attribute syntax must produce octet strings. To the greatest extent possible, encoded octet strings should be usable in their native encoded form for display purposes. In particular, encoding rules for attribute syntaxes defining non-binary values should produce strings that can be displayed with little or no translation by clients implementing the lightweight directory protocols.

4. Standard Attribute Syntax Encodings

For the purposes of defining the encoding rules for the standard attribute syntaxes, the following auxiliary BNF definitions will be used:

```
<a>  ::= 'a'  | 'b'  | 'c'  | 'd'  | 'e'  | 'f'  | 'g'  | 'h'  | 'i'  |
         'j'  | 'k'  | 'l'  | 'm'  | 'n'  | 'o'  | 'p'  | 'q'  | 'r'  |
         's'  | 't'  | 'u'  | 'v'  | 'w'  | 'x'  | 'y'  | 'z'  | 'A'  |
         'B'  | 'C'  | 'D'  | 'E'  | 'F'  | 'G'  | 'H'  | 'I'  | 'J'  |
         'K'  | 'L'  | 'M'  | 'N'  | 'O'  | 'P'  | 'Q'  | 'R'  | 'S'  |
         'T'  | 'U'  | 'V'  | 'W'  | 'X'  | 'Y'  | 'Z'  |

<d> ::= '0'  | '1'  | '2'  | '3'  | '4'  | '5'  | '6'  | '7'  | '8'  |'9'

<hex-digit>    ::= <d>| 'a'  | 'b'  | 'c'  | 'd'  | 'e'  | 'f'  |
                 'A'  | 'B'  | 'C'  | 'D'  | 'E'  | 'F'

<k> ::= <a> | '-'

<p> ::= <a>   | <d> |  '''  | '('  | ')'  | '+'  | ','  | '-'  | '-'  | '.' |
        ' /'  | ':' | '?'  | ' '

<CRLF> ::= The ASCII newline character with hexadecimal value OxOA

<letterstring> ::=      <a>   |     <a> <letterstring>

<numericstring> ::=     <d>   |     <d> <numericstring>

<keystring> ::=         <k>   |     <k> <keystring>

<printablestring> ::=   <p>   |     <p> <printablestring>
```

```
<space> ::= ' '  | ' ' <space>
```

4.1 . Undefined

Values of type Undefined are encoded as if they were values of type Octet String.

4.2. Case Ignore String

A string of type caseIgnoreStringSyntax is encoded as the string value itself.

4.3. Case Exact String

The encoding of a string of type caseExactStringSyntax is the string value itself.

4.4. Printable String

The encoding of a string of type printableStringSyntax is the string value itself.

4.5. Numeric String

The encoding of a string of type numericStringSyntax is the string value itself.

4.6. Octet String

The encoding of a string of type octetStringSyntax is the string value itself.

4.7. Case Ignore IA5 String

The encoding of a string of type caseIgnoreIA5String is the string value itself.

4.8. IA5 String

The encoding of a string of type iA5StringSyntax is the string value itself.

4.9. T61 String

The encoding of a string of type t61StringSyntax is the string value itself.

4.10. Case Ignore List

Values of type caseIgnoreListSyntax are encoded according to the following BNF:

```
<caseignorelist> ::= <caseignorestring> |
                     <caseignorestring> '$' <caseignorelist>

<caseignorestring> ::= a string encoded according to the rules for Case
                       Ignore String as above.
```

4.11. Case Exact List

Values of type caseExactListSyntax are encoded according to the following BNF:

```
<caseexactlist> ::= <caseexactstring> |
                    <caseexactstring> '$' <caseexactlist>

<caseexactstring> ::= a string encoded according to the rules for Case
                      Exact String as above.
```

4.12. Distinguished Name

Values of type distinguishedNameSyntax are encoded to have the representation defined in [5].

4.13. Boolean

Values of type booleanSyntax are encoded according to the following BNF:

```
<boolean> ::= "TRUE" | "FALSE"
```

Boolean values have an encoding of "TRUE" if they are logically true, and have an encoding of "FALSE" otherwise.

4.14. Integer

Values of type IntegerSyntax are encoded as the decimal representation of their values, with each decimal digit represented by the its character equlvalent. So the digit 1 is represented by the character '1', the digit 2 is represented by the character '2' and so on.

4.15. Object Identifier

Values of type objectIdentifierSyntax are encoded according to the following BNF:

```
<oid> ::= <descr> | <descr> '.' <numericoid> | <numericoid>

<descr> ::= <keystring>

<numericoid> ::= <numericstring> | cnumericstring> '.' <numericoid>
```

In the above BNF, <descr> is the syntactic representation of an object descriptor. When encoding values of type objectIdentifierSyntax, the first encoding option should be used in preference to the second, which should be used in preference to the third wherever possible. That is, in encoding object identifiers, object descriptors (where assigned and known by the implementation) should be used in preference to numeric oids to the greatest extent possible. For example, in encoding the object identifier representing an organizationName, the descriptor "organizationName" is preferable to "ds.4.10", which is in turn preferable to the string "2.5.4.10".

4.16. Telephone Number

Values of type telephoneNumberSyntax are encoded as if they were Printable String types.

4.17. Telex Number

Values of type telexNumberSyntax are encoded according to the following BNF:

```
<telex-number> ::= <actual-number> '$' <country> '$' <answerback>

<actual-number> ::= <printablestring>

<country> ::= <printablestring>

<answerback> ::= <printablestring>
```

In the above, <actual-number> is the syntactic representation of the number portion of the TELEX number being encoded, <country> is the TELEX country code, and <answerback> is the answerback code of a TELEX terminal.

4.18. Teletex Terminal Identifier

Values of type teletexTerminalIdentifier are encoded according to the following BNF:

```
<teletex-id> ::= <printablestring>  0*( '$' <printablestring>)
```

In the above, the first <printablestring> is the encoding of the first portion of the teletex terminal identifier to be encoded, and the subsequent 0 or more <printablestrings> are subsequent portions of the teletex terminal identifier.

4.19. Facsimile Telephone Number

Values of type FacsimileTelephoneNumber are encoded according to the following BNF:

```
<fax-number> ::= <printablestring> [ '$' <faxparameters> ]

<faxparameters> ::= <faxparm> | <faxparm> '$' <faxparameters>

<faxparm> ::= 'twoDimensional'  | 'fineResolution'  | 'unlimitedLength' |
              'b4Length'  | 'a3Width' | 'b4Width'  | 'uncompressed'
```

In the above, the first <printablestring> is the actual fax number, and the <faxparm> tokens represent fax parameters.

4.20. Presentation Address

Values of type PresentationAddress are encoded to have the representation described in [6].

4.21. UTC Time

Values of type uTCTimeSyntax are encoded as if they were Printable Strings with the strings containing a UTCTime value.

4.22. Guide (search guide)

Values of type Guide, such as values of the searchGuide attribute, are encoded according to the following BNF:

```
<guide-value> ::= [ <object-class> '#' ] <criteria>

<object-class> ::= an encoded value of type objectIdentifierSyntax

<criteria> ::= <criteria-item> | <criteria-set> | '!' <criteria>

<criteria-set> ::= [ '(' ] <criteria> '&' <criteria-set> [ ')' ] |
                   [ '(' ] <criteria>  '|'  <criteria-set> [ ')' ]

<criteria-item> ::= [ '(' ] <attributetype> '$' <match-type> [ ')' ]

<match-type> ::= "EQ" | "SUBSTR" | "GE" | "LE" | "APPROX"
```

4.23. Postal Address

Values of type PostalAddress are encoded according to the following BNF:

```
<postal-address> ::= <t61string> | <t61string> '$' <postal-address>
```

In the above, each <t61string> component of a postal address value is encoded as a value of type t61StringSyntax.

4.24. User Password

Values of type userPasswordSyntax are encoded as if they were of type octetStringSyntax.

4.25. User Certificate

Values of type userCertificate are encoded according to the following BNF:

```
<certificate> ::= <signature> '#' <issuer> '#' <validity> '~' <subject>
                  '#' <public-key-info>

<signature> ::= <algorithm-id>

<issuer> ::= an encoded Distinguished Name

<validity> ::= <not-before-time> '#' <not-after-time>

<not-before-time> ::= <utc-time>

<not-after-time> ::= <utc-time>

<algorithm-parameters> ::= <null> | <integervalue> |
                           '{ASN}' <hex-string>

<subject> ::= an encoded Distinguished Name

<public-key-info> ::= <algorithm-id> '#' <encrypted-value>

<encrypted-value> ::= <hex-string> | <hex-string> '-' <d>

<algorithm-id> ::= <oid> '#' <algorithm-parameters>

<utc-time> ::= an encoded UTCTime value

<hex-string> ::= <hex-digit> | <hex-digit> <hex-string>
```

4.26. CA Certificate

Values of type cACertificate are encoded as if the values were of type userCertificate.

4.27. Authority Revocation List

Values of type authorityRevocationList are encoded according to the following BNF:

```
<certificate-list> ::= <signature> '#' <issuer> '#'
                       <utc-time> [ '#' <revoked-certificates> ]

<revoked-certificates> ::= <algorithm> '#' <encrypted-value>
                           [ '#' 0*(<revoked-certificate>) '#']

<revoked-certificates> ::= <subject> '#' <algorithm> '#'
                           <serial> '#' <utc-time>
```

The syntactic components <algorithm>, <issuer>, <encrypted-value>, <utc-time>, <subject> and <serial> have the same definitions as in the BNF for the userCertificate attribute syntax.

4.28. Certificate Revocation List

Values of type certificateRevocationList are encoded as if the values were of type authorityRevocationList.

4.29. Cross Certificate Pair

Values of type crossCertificatePair are encoded according to the following BNF:

```
<certificate-pair> ::= <certificate> '|' <certificate>
```

The syntactic component <certificate> has the same definition as in the BNF for the userCertificate attribute syntax.

4.30. Delivery Method

Values of type deliveryMethod are encoded according to the following BNF:

```
<delivery-value> ::= <pdm> | <pdm> '$' <delivery-value>

<pdm> ::= 'any' | 'mhs' | 'physical' | 'telex' | 'teletex' |
          'g3fax'  | 'g4fax'  | 'ia5'  | 'videotex'  | 'telephone'
```

4.31. Other Mailbox

Values of the type otherMailboxSyntax are encoded according to the following BNF:

```
<otherMailbox> ::= <mailbox-type> '$' <mailbox>

<mailbox-type> ::= an encoded Printable String

<mailbox> ::= an encoded IA5 String
```

In the above, <mailbox-type> represents the type of mail system in which the mailbox resides, for example "Internet" or "MCIMail"; and <mailbox> is the actual mailbox in the mail system defined by <mailbox-type>.

4.32. Mail Preference

Values of type mailPreferenceOption are encoded according to the following BNF:

```
<mail-preference> ::= "NO-LISTS" | "ANY-LIST" | "PROFESSIONAL-LISTS"
```

4.33. Photo

Values of type Photo are encoded as if they were octet strings containing JPEG images in the JPEG File Interchange Format (JFIF).

4.34. Fax

Values of type Fax are encoded as if they were octet strings containing Group 3 Fax images.

5. Security Considerations

Security considerations are not discussed in this document.

6. Bibliography

[1] The Directory: Selected Attribute Syntaxes
CCITT; Recommendation X.520

[2] Information Processing Systems -- Open Systems Interconnection --
The Directory: Selected Attribute Syntaxes

[3] The COSINE and Internet X.500 Schema
Paul Barker, Steve Hardcastle-Kille; Request for Comment (RFC) 1274

[4] The ISO Development Environment: User's Manual -- Volume 5: QUIPU
Colin Robbins, Stephen E. Hardcastle-Kille

[5] A String Representation of Distinguished Names
Steve Hardcastle-Kille; OSI-DS document 23

[6] A String Representation for Presentation Addresses
Steve Hardcastle-Kille; Request for Comment (RFC) 1278

Appendix E

Lightweight Directory Access Protocol

Appendix E

Network Working Group
INTERNET-DRAFT

Wengyik Yeong
Performance Systems International
Tim Howes
University of Michigan
Steve Hardcastle-Kille
University College London

Lightweight Directory Access Protocol

1. Status of this Memo

This draft document will be submitted to the RFC Editor as a standards document. Distribution of this memo is unlimited. Please send comments to the authors, or the discussion group <osi-ds@cs.ucl.ac.uk>.

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts).

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a "working draft" or "work in progress."

Please check the I-D abstract listing contained in each Internet Draft directory to learn the current status of this or any other Internet Draft.

2. Abstract

The tremendous interest in X.500 [1,2] technology in the Internet has lead to efforts to reduce the high ``cost of entry'' associated with use of the technology, such as the Directory Assistance Service [3] and DIXIE [4]. While efforts such as these have met with success, they have been solutions based on particular implementations and as such have limited applicability. This document continues the efforts to define Directory protocol alternatives but departs from previous efforts in that it consciously avoids dependence on particular implementations.

The protocol described in this document is the first of a series of protocols designed to provide access to the Directory while not incurring the resource requirements of the Directory Access Protocol (DAP). This protocol is specifically targeted at simple management applications and

browser applications that provide simple read/write interactive access to the Directory, and is intended to be a complement to the DAP itself.

3. Protocol Model

The general model adopted by this protocol is one of clients performing protocol operations against servers. In this model, this is accomplished by a client transmitting a protocol request describing the operation to be performed to a server, which is then responsible for performing the necessary operations on the Directory. Upon completion of the necessary operations, the server returns a response containing any results or errors to the requesting client. In keeping with the goal of easing the costs associated with use of the Directory, it is an objective of this protocol to minimize the complexity of clients so as to facilitate widespread deployment of applications capable of utilizing the Directory.

Note that, although servers are required to return responses whenever such responses are defined in the protocol, there is no requirement for synchronous behavior on the part of either client or server implementations: requests and responses for multiple operations may be exchanged by client and servers in any order, as long as clients eventually receive a response for every request that requires one.

Consistent with the model of servers performing protocol operations on behalf of clients, it is also to be noted that protocol servers are expected to handle referrals without resorting to the return of such referrals to the client. This protocol makes no provisions for the return of referrals to clients, as the model is one of servers ensuring the performance of all necessary operations in the Directory, with only final results or errors being returned by servers to clients.

This protocol is designed to run over connection-oriented, reliable transports, with all 8 bits in an octet being significant in the data stream. Server implementations running over the TCP should provide a protocol listener on port 389.

4. Elements of Protocol

For the purposes of protocol exchanges, all protocol operations are encapsulated in a common envelope, the LDAPMessage, which is defined as follows:

```
LDAPMessage ::=
    SEQUENCE {
         messageID      MessageID,
         protocolOp     CHOICE {
                             bindRequest         BindRequest,
                                  bindResponse         BindResponse,
                                  unbindRequest        UnbindRequest
                                  searchRequest        SearchRequest,
                                  searchResponse       SearchResponse,
                                  modifyRequest        ModifyRequest,
                                  modifyResponse       ModifyResponse,
                                  addRequest           AddRequest,
                                  addResponse          AddResponse,
                                  delRequest           DelRequest,
                                  delResponse          DelResponse,
                                  modifyRDNRequest     ModifyRDNRequest,
                                  modifyRDNResponse    ModifyRDNResponse,
                                  compareDNRequest     CompareRequest,
                                  compareDNResponse    CompareResponse,
                                  abandonRequest       AbandonRequest
```

```
     MessageID ::= INTEGER (0 .. MaxInt)
```

The function of the LDAPMessage is to provide a envelope containing common fields required in all protocol exchanges. At this time the only common field is a message ID, which is required to have a value different from the values of any other requests outstanding in the LDAP session of which this message is a part.

The message ID value must be echoed in all LDAPMessage envelopes encapsulting responses corresponding to the request contained in the LDAPMessage in which the message ID value was originally used.

In addition to the LDAPMessage defined above, the following definitions are also used in defining protocol operations:

```
     IA5String ::= OCTET STRING
```

The IA5String is a notational convenience to indicate that, although strings of IA5String type encode as OCTET STRING types, the legal character set in such strings is limited to the IA5 character set.

```
     LDAPDN ::= IA5String

     RelativeLDAPDN ::= IA5String
```

An LDAPDN and a RelativeLDAPDN are respectively defined to be the representation of a Distinguished Name and a Relative Distinguished Name after encoding according to the specification in [5], such that

```
     <distinguished-name> ::= <name>

     <relative-distinguished-name> ::= <name-component>
```

where <name> and <name-component> are as defined in [5].

```
     AttributeValueAssertion ::=
         SEQUENCE {
              attributeType        AttributeType
              attributeValue       AttributeValue
         }
```

The AttributeValueAssertion type definition is similar to the one in the Directory standards.

```
     AttributeType ::= IA5String

     AttributeValue ::= OCTET STRING
```

An AttributeType value takes on as its value the textual string associated with that AttributeType in the Directory standards. For example, the AttributeType 'organizationName' with object identifier 2.5.4.10 is represented as an AttributeType in this protocol by the string ``organizationName''. In the event that a protocol implementation encounters an Attribute Type with which it cannot associate a textual string, an ASCII string encoding of the object identifier associated with the Attribute Type may be subsitituted. For example, the organizationName AttributeType may be represented by the ASCII string "2.5.4.10" if a protocol implementation is unable to associate the string ``organizationName'' with it.

A field of type AttributeValue takes on as its value an octet string encoding of a Directory AttributeValue type. The definition of these string encodings for different Directory AttributeValue types may be found in companions to this document that define the encodings of various attribute syntaxes such as [6].

```
     LDAPResult ::=
         SEQUENCE {
             resultCode    ENUMERATED {
                             success                    (0),
                             operationsError            (1),
                             protocolError              (2),
                             timeLimitExceeded          (3),
                             sizeLimitExceeded          (4),
                             compareFalse               (5),
                             compareTrue                (6),
                                authMethodNotSupported       (7),
                                strongAuthRequired           (8),
                                noSuchAttribute              (16),
                                invalidAttributeSyntax       (17),
                                undefinedAttributeType       (18),
                                inappropriateMatching        (19),
                                constraintViolation          (20),
                                attributeOrValueExists       (21),
                                noSuchObject                 (32),
                                aliasProblem                 (33),
                                invalidAttributeSyntax       (34),
                                isLeaf                       (35),
                                aliasDereferencingProblem    (36),
                                inappropriateAuthentication  (48),
                                invalidCredentials           (49),
                                insufficientAccessRights     (50),
                                busy                         (51),
                                unavailable                  (52),
                                unwillingToPerform           (53),
                                loopDetect                   (54),
                                namingViolation              (64),
                                objectClassViolation         (65),
                                notAllowedOnNonLeaf          (66),
                                notAllowedOnRDN              (67),
                                entryAlreadyExists           (68),
                                objectClassModsProhibited    (69),
                                other                        (80)
                             },
                errorMessage  IA5String
        }
```

The LDAPResult is the construct used in this protocol to return success or failure indications from servers to clients. In response to various requests, servers will return responses containing fields of type LDAPResult to indicate the final status of a protocol operation request. The errorMessage field of this construct may, at the servers option, be used to return an ASCII string containing a textual, human-readable error diagnostic. If the server chooses not to return a textual diagnostic, the errorMessage field of the LDAPResult type should contain a zero length string.

4.1. Bind Operation

The function of the Bind Operation is to initiate a protocol session between a client and a server, and to allow the authentication of the client to the server. The Bind Operation must be the first operation request received by a server from a client in a protocol session. The Bind Request is defined as follows:

```
     BindRequest ::=
         [APPLICATION 0] SEQUENCE {
                         version   INTEGER (1 .. 127),
                         name      LDAPDN,
                         authentication CHOICE {
                              simple        [0] OCTET STRING
                              krbv42LDAP    [1] OCTET STRING,
                              krbv42DSA     [2] OCTET STRING
                         }
         }
```

Parameters of the Bind Request are:

- version: A version number indicating the version of the protocol to be used in this protocol session. Currently, the only legal value of this field is the numeric value 1.

- name: The name of the Directory object that the client wishes to bind as. This field may take on a null value for the purposes of anonymous binds.

- authentication: information used to authenticate the name, if any, provided in the Bind Request. The ``simple'' authentication option provides minimal authentication facilities, with the contents of the authentication field consisting only of a cleartext password. This option should be used when unauthenticated or anonymous binds are to be performed, with the field containing a zero length string in such cases. In instances when better authentication facilities are required, the 'krbv42LDAP' and 'krbv42DSA' options should be used, with the fields containing values as returned by the KERBEROS [7] krb_mk_req() calls. Kerberos version 4 authentication to the LDAP server and the DSA is accomplished by using the ``krbv42LDAP'' and ``krbv42DSA'' authentication options, respectively. Each octet string should contain the kerberos ticket (e.g., as returned by krb_mk_req()) for the appropriate service. The suggested service name for authentication to the LDAP server is "ldapserver". The suggested service name for authentication to the DSA is "x500dsa". In both cases, the suggested instance name for the service is the name of the host on which the service is running. Of course, the actual service names and instances will depend on what is entered in the local kerberos principle database.

The Bind Operation requires a response, the Bind Response, which is defined as:

```
     BindResponse ::= [APPLICATION 1] LDAPResult
```

A Bind Response consists simply of an indication from the server of the status of the client's request for the initiation of a protocol session.

Upon receipt of a Bind Request, a protocol server will authenticate the requesting client if necessary, and attempt to set up a protocol session with that client. The server will then return a Bind Response to the client indicating the status of the session setup request.

4.2. Unbind Operation

The function of the Unbind Operation is to terminate a protocol session. The Unbind Operation is defined as follows:

```
     UnbindRequest ::= [APPLICATION 2] NULL
```

The Unbind Operation has no response defined. Upon transmission of an UnbindRequest, a protocol client may assume that the protocol session is terminated. Upon receipt of an UnbindRequest, a protocol server may assume that the requesting client has terminated the session and that all outstanding requests may be discarded.

4.3. Search Operation

The Search Operation allows a client to request that a search be performed on its behalf by a server. The Search Request is defined as follows:

```
     SearchRequest ::=
         [APPLICATION 3] SEQUENCE {
             baseObject    LDAPDN,
             scope         ENUMERATED {
                                baseObject              (0),
                                singleLevel             (1),
                                wholeSubtree            (2)
                           },
             derefAliases  ENUMERATED {
                                        neverDerefAliases       (0),
                                        derefInSearching        (1),
                                        derefFindingBaseObj     (2),
                                        derefAlways             (3)
                                   },
             sizeLimit     INTEGER (0 .. MaxInt),
             timeLimit     INTEGER (0 .. MaxInt),
             attrsOnly     BOOLEAN,
             filter        Filter,
             attributes    SEQUENCE OF AttributeType
     }

     Filter ::=
          CHOICE {
              and                 [0] SET OF Filter,
              or                  [1] SET OF Filter,
              not                 [2] Filter,
              equalityMatch       [3] AttributeValueAssertion,
              substrings          [4] SubstringFilter,
              greaterOrEqual      [5] AttributeValueAssertion,
              lessOrEqual         [6] AttributeValueAssertion,
              present             [7] AttributeType,
              approxMatch         [8] AttributeValueAssertion
          }

      SubstringFilter
          SEQUENCE {
              type                AttributeType,
              SEQUENCE OF CHOICE {
                initial           [0] IA5String,
                any               [1] IA5String,
                final             [2] IA5String
              }
          }
```

Parameters of the Search Request are:

- baseObject: A LDAPDN that is the base object entry relative to which the search is to be performed.

- scope: An indicator of the scope of the search to be performed. The semantics of the possible values of this field are identical to the semantics of the scope field in the Directory Search Operation.

- derefAliases: An indicator as to how alias objects should be handled in searching. The semantics of the possible values of this field are, in order of increasing value:-

  neverDerefAliases: don't dereference aliases in searching or in locating the base object of the search;

  derefInSearching: dereference aliases in subordinates of the base object in searching, but not in locating the base object of the search;

  derefFindingBaseObject: dereference aliases in locating the base object of the search, but not when searching subordinates of the base object;

  derefAlways: dereference aliases both in searching and in locating the base object of the search.

- sizelimit: A sizelimit that restricts the maximum number of entries to be returned as a result of the search. A value of 0 in this field-indicates that no sizelimit restrictions are in effect for the search.

- timelimit: A timelimit that restricts the maximum time (in seconds allowed for a search). A value of 0 in this field indicates that no timelimit restrictions are in effect for the search.

- attrsOnly: An indicator as to whether search results should contain both attribute types and values, or just attribute types.

- filter: A filter that defines the conditions that must be fulfilled in order for the search to match a given entry.

- attributes: A list of the attributes from each entry found as a result of the search to be returned. An empty list signifies that all attributes from each entry found in the search are to be returned.

The results of the search attempted by the server upon receipt of a Search Request are returned in Search Responses, defined as follows:

```
Search Response ::=
    CHOICE {
         entry          [APPLICATION 4] SEQUENCE {
                              objectName     LDAPDN,
                              attributes     SEQUENCE OF SEQUENCE {
                                                  AttributeType,
                                                  SET OF AttributeValue
                                             }
                        },
         resultCode     [APPLICATION 5] LDAPResult
    }
```

Upon receipt of a Search Request, a server will perform the necessary search of the DIT.

The server will return to the client a sequence of responses comprised of:

- Zero or more Search Responses each consisting of an entry found during the search; with the response sequence terminated by

- A single Search Response containing an indication of success, or detailing any errors that have occurred.

Each entry returned will contain all attributes, complete with associated values if necessary, as specified in the 'attributes' field of the Search Request.

4.4. Modify Operation

The Modify Operation allows a client to request that a modification of the DIB be performed on its behalf by a server. The Modify Request is defined as follows:

```
     ModifyRequest ::=
         [APPLICATION 6] SEQUENCE {
              object         LDAPDN,
              modification   SEQUENCE OF SEQUENCE {
                                  operation      ENUMERATED {
                                                      add       (0),
                                                      delete    (1),
                                                      replace   (2)
                                                 },
                                  modification   SEQUENCE {
                                                    type     AttributeType,
                                                    values   SET OF
                                                               AttributeValue
                                                 }
                             }
         }
```

Parameters of the Modify Request are:

- object: The object to be modified. The Distinguished Name value of this field should name the object to be modified after all aliases have been dereferenced. The server will not perform any alias dereferencing in determining the object to be modified.

- A list of modifications to be performed on the entry to be modified. The entire list of entry modifications should be performed in the order they are listed, as a single atomic operation. While individual modifications may violate the Directory schema, the resulting entry after the entire list of modifications is performed must conform to the requirements of the Directory schema. The values that may be taken on by the 'operation' field in each modification construct have the following semantics respectively:-

  add: add values listed to the given attribute, creating the attribute if necessary;

  delete: delete values listed from the given attribute, removing the entire attribute if no values are listed;

replace: replace existing values of the given attribute with the new values listed, creating the attribute if necessary.

The result of the modify attempted by the server upon receipt of a Modify Request is returned in a Modify Response, defined as follows:

```
ModifyResponse ::= [APPLICATION 7] LDAPResult
```

Upon receipt of a Modify Request, a server will perform the necessary modifications to the DIB.

The server will return to the client a single Modify Response indicating either the successful completion of the DIB modification, or the reason that the modification failed. Note that due to the requirement for atomicity in applying tHe list of modifications in the Modify Request, the client may expect that no modifications of the DIB have been performed if the Modify Response received indicates any sort of error, and that all requested modifications have been performed if the Modify Response indicates successful completion of the Modify Operation.

4.5. Add Operation

The Add Operation allows a client to request the addition of an entry into the Directory. The Add Request is defined as follows:

```
AddRequest ::=
    [APPLICATION 8] SEQUENCE {
         entry          LDAPDN,
         attrs          SEQUENCE OF SEQUENCE {
                             type          AttributeType,
                             values        SET OF AttributeValue
                        }
    }
```

Parameters of the Add Request are:

- entry: the Distinguished Name of the entry to be added. Note that all components of the name except for the last RDN component must exist for the add to succeed.

- attrs: the list of attributes that make up the content of the entry being added.

The result of the add attempted by the server upon receipt of a Add Request is returned in the Add Response, defined as follows:

```
     AddResponse ::= [APPLICATION 9] LDAPResult
```

Upon receipt of an Add Request, a server will attempt to perform the add requested. The result of the add attempt will be returned to the client in the Add Response.

4.6. Delete Operation

The Delete Operation allows a client to request the removal of an entry from the Directory. The Delete Request is defined as follows:

```
     DelRequest ::= [APPLICATION 10] LDAPDN
```

The Delete Request consists only of the Distinguished Name of the entry to be deleted. The result of the delete attempted by the server upon receipt of a Delete Request is returned in the Delete Response, defined as follows:

```
     DelResponse ::= [APPLICATION 11] LDAPResult
```

Upon receipt of a Delete Request, a server will attempt to perform the entry removal requested. The result of the delete attempt will be returned to the client in the Delete Response. Note that only leaf objects may be deleted with this operation.

4.7. Modify RDN Operation

The Modify RDN Operation allows a client to change the last component of the name of an entry in the Directory. The Modify RDN Request is defined as follows:

```
     ModifyRDNRequest ::=
         [APPLICATION 12] SEQUENCE {
              entry          LDAPDN,
              newrdn         RelativeLDAPDN
         }
```

Parameters of the Modify RDN Request are:

- entry: the name of the entry to be changed.

- newrdn: the RDN that will form the last component of the new name.

The result of the name change attempted by the server upon receipt of a Modify RDN Request is returned in the Modify RDN Response, defined as follows:

```
     ModifyRDNResponse ::= [APPLICATION 13] LDAPResult
```

Upon receipt of a Modify RDN Request, a server will attempt to perform the name change. The result of the name change attempt will be returned to the client in the Modify RDN Response. The attributes that make up the old RDN are not deleted from the entry.

4.8. Compare Operation

The Compare Operation allows a client to compare an assertion provided with an entry in the Directory. The Compare Request is defined as follows:

```
     CompareRequest ::=
         [APPLICATION 14] SEQUENCE {
              entry          LDAPDN,
              ava            AttributeValueAssertion
         }
```

Parameters of the Compare Request are:

- entry: the name of the entry to be compared with.
- ava: the assertion with which the entry is to be compared.

The result of the compare attempted by the server upon receipt of a Compare Request is returned in the Compare Response, defined as follows:

```
     CompareResponse ::= [APPLICATION 15] LDAPResult
```

Upon receipt of a Compare Request, a server will attempt to perform the requested comparison. The result of the comparison will be returned to the client in the Compare Response. Note that errors and the result of comparison are all returned in the same construct.

4.9. Abandon Operation

The function of the Abandon Operation is to allow a client to request that the server abandon an outstanding operation. The Abandon Request is defined as follows:

```
     AbandonRequest ::= [APPLICATION 16] MessageID
```

There is no response defined in the Abandon Operation. Upon transmission of an Abandon Operation, a client may expect that the operation identified by the Message ID in the Abandon Request has been abandoned. In the event that a server receives an Abandon Request on a Search Operation in the midst of transmitting responses to that search, that server should cease transmitting responses to the abandoned search immediately.

5. Protocol Element Encodings

The protocol elements of LDAP are encoded for exchange using the Basic Encoding Rules (BER) of ASN.1. However, due to the high overhead involved in using certain elements of the BER, the following additional restrictions are placed on BER-encodings of LDAP protocol elements:

(1) Only the definite form of length encoding will be used.

(2) Bitstrings and octet strings will be encoded in the primitive form only.

6. Security Considerations

This version of the protocol provides facilities only for simple authentication using a cleartext password.

7. Bibliography

[1] The Directory: Overview of Concepts, Models and Service
CCITT; Recommendation X.500, 1988

[2] Information Processing Systems -- Open Systems Interconnection --
The Directory: Overview of Concepts, Models and Service
ISO/IEC JTC 1/SC21; International Standard 9594-1, 1988

[3] Directory Assistance Service
M.T. Rose; RFC 1202, February 1991.

[4] DIXIE protocol specification
T. Howes, M. Smith, B. Beecher; RFC1249, August 1991.

[5] A String Representation of Distinguished Names
Steve Hardcastle-Kille; OSI-DS document 23

[6] The String Representation of Standard Attribute Syntaxes
T. Howes, S. Hardcastle-Kille, W. Yeong, C.J. Robbins; OSI-DS document 27

[7] Kerberos Authentication and Authorization System
S.P. Miller, B.C. Neuman, J.I. Schiller, J.H. Saltzer; MIT Project Athena Documentation Section E.2.1, December 1987

[8] The Directory: Models
CCITT; Recommendation X.501
ISO/IEC JTC 1/SC21; International Standard 9594-2, 1988

[10] The Directory: Abstract Service Definition

CCITT; Recommendation X.511
ISO/IEC JTC 1/SC21; International Standard 9594-3, 1988

Appendix A
Complete ASN.1 Definition

```
Lightweight-Directory-Access-Protocol DEFINITIONS ::=

IMPLICIT TAGS

BEGIN

LDAPMessage ::=
    SEQUENCE {
      messageID      MessageID,
     -- unique id in request,
     -- to be echoed in response(s)
      protocolOp     CHOICE {
          searchRequest       SearchRequest,
          searchResponse      SearchResponse,
          modifyRequest       ModifyRequest,
          modifyResponse      ModifyResponse,
          addRequest AddRequest,
          addResponse AddResponse,
          delRequest DelRequest,
          delResponse DelResponse,
          modifyDNRequest ModifyDNRequest,
          modifyDNResponse ModifyDNResponse,
          compareDNRequest CompareRequest,
          compareDNResponse CompareResponse,
          bindRequest         BindRequest,
          bindResponse        BindResponse,
          abandonRequest      AbandonRequest,
          unbindRequest       UnbindRequest
     }
    }

BindRequest ::=
    [APPLICATION 0] SEQUENCE {
      version        INTEGER (1 .. 127),
     -- current version is 1
      name           LDAPDN,
     -- null name implies an anonymous bind
         authentication CHOICE {
                             simple        [0] OCTET STRING,
                    -- a zero length octet string
                    -- implies an unauthenticated
                    -- bind.
                             krbv42LDAP    [1] OCTET STRING,
                             krbv42DSA     [2] OCTET STRING
                    -- values as returned by krb_mk_req()
                    -- Other values in later versions
                    -- of this protocol.
                        }
    }

BindResponse ::= [APPLICATION 1] LDAPResult

UnbindRequest ::= [APPLICATION 2] NULL

SearchRequest ::=
    [APPLICATION 3] SEQUENCE {
      baseObject     LDAPDN,
      scope          ENUMERATED {
          baseObject             (0),
          singleLevel            (1),
          wholeSubtree           (2)
     },
      derefAliases   ENUMERATED {
                             neverDerefAliases      (0),
                             derefInSearching       (1),
                             derefFindingBaseObj    (2),
                             alwaysDerefAliases     (3)
                        },
         sizeLimit      INTEGER (0 .. MaxInt),
                        -- value of 0 implies no sizelimit
         timeLimit      INTEGER (0 .. MaxInt),
                        -- value of 0 implies no timelimit
         attrsOnly     BOOLEAN,
                        -- TRUE, if only attributes (without values)
                        -- to be returned.
         filter         Filter,
         attributes     SEQUENCE OF AttributeType
    }

SearchResponse ::=
    CHOICE {
         entry          [APPLICATION 4] SEQUENCE {
                             objectName     LDAPDN,
                             attributes     SEQUENCE OF SEQUENCE {
                                              AttributeType,
                                              SET OF
                                                AttributeValue
                                            }
                        },
         resultCode     [APPLICATION 5] LDAPResult
    ModifyRequest ::=
        [APPLICATION 6] SEQUENCE {
             object          LDAPDN,
             modifications   SEQUENCE OF SEQUENCE {
                                  operation     ENUMERATED {
                                                  add       (0),
                                                  delete    (1),
                                                  replace   (2)
                                                },
                                  modification  SEQUENCE {
                                                  type      AttributeType
                                                  values    SET OF
                                                              AttributeValue
                                                }
                             }
        }

    ModifyResponse ::= [APPLICATION 7] LDAPResult

    AddRequest ::=
        [APPLICATION 8] SEQUENCE {
             entry           LDAPDN,
             attrs           SEQUENCE OF SEQUENCE {
                                  type          AttributeType,
                                  values        SET OF AttributeValue
                             }
        }

    AddResponse ::= [APPLICATION 9] LDAPResult

    DelRequest ::= [APPLICATION 10] LDAPDN

    DelResponse ::= [APPLICATION 11] LDAPResult

    ModifyRDNRequest ::=
        [APPLICATION 12] SEQUENCE {
             entry           LDAPDN,
             newrdn          RelativeLDAPDN -- old RDN always deleted
        }

    ModifyRDNResponse ::= [APPLICATION 13] LDAPResult

    CompareRequest ::=
        [APPLICATION 14] SEQUENCE {
             entry           LDAPDN,
             ava             AttributeValueAssertion
        }
```

```
CompareResponse ::= [APPLICATION 15] LDAPResult

AbandonRequest ::= [APPLICATION 16] MessageID

MessageID ::= INTEGER (0 .. MaxInt)

LDAPDN ::= IA5String

RelativeLDAPDN ::= IA5String

Filter ::=
    CHOICE {
        and             [0] SET OF Filter,
        or              [1] SET OF Filter,
        not             [2] Filter,
        equalityMatch   [3] AttributeValueAssertion,
        substrings      [4] SubstringFilter,
        greaterOrEqual  [5] AttributeValueAssertion,
        lessOrEqual     [6] AttributeValueAssertion,
        present         [7] AttributeType,
        approxMatch     [8] AttributeValueAssertion
    }

LDAPResult ::=
    SEQUENCE {
        resultCode    ENUMERATED {
                        success                      (0),
                        operationsError              (1),
                        protocolError                (2),
                        timeLimitExceeded            (3),
                        sizeLimitExceeded            (4),
                        compareFalse                 (5),
                        compareTrue                  (6),
                        authMethodNotSupported       (7),
                        strongAuthRequired           (8),
                        noSuchAttribute              (16),
                        invalidAttributeSyntax       (17),
                        undefinedAttributeType       (18),
                        inappropriateMatching        (19),
                        constraintViolation          (20),
                        attributeOrValueExists       (21),
                        noSuchObject                 (32),
                        aliasProblem                 (33),
                        invalidAttributeSyntax       (34),
                        isLeaf                       (35),
                        aliasDereferencingProblem    (36),
                        inappropriateAuthentication  (48),
                        invalidCredentials           (49),
                                insufficientAccessRights     (50),
                                busy                         (51),
                                unavailable                  (52),
                                unwillingToPerform           (53),
                                loopDetect                   (54),
                                namingViolation              (64),
                                objectClassViolation         (65),
                                notAllowedOnNonLeaf          (66),
                                notAllowedOnRDN              (67),
                                entryAlreadyExists           (68),
                                objectClassModsProhibited    (69),
                                other                        (80)
                              },
             errorMessage  IA5String
         }

     AttributeType ::= IA5String
                     -- text name of the attribute, or dotted
                     -- OID representation

     AttributeValue ::= OCTET STRING

     AttributeValueAssertion ::=
         SEQUENCE {
             attributeType        AttributeType,
             attributeValue       AttributeValue
         }

     SubstringFilter
         SEQUENCE {
             type               AttributeType,
             SEQUENCE OF CHOICE {
            initial          [0] IA5String,
            any              [1] IA5String,
            final            [2] IA5String
           }
         }

     IA5String ::= OCTET STRING

     MaxInt ::= 65535
     END
```

Appendix F

Comments on NADF Agreements For Name and Knowledge Sharing

Comments on NADF Agreements for Name and Knowledge Sharing, by Wengyik Yeong, Performance Systems International, Inc., yeongw@psi.com, March 5, 1992.

1. Introduction
After a review of the NADF Agreements on Name and Knowledge Sharing documented in DF 277, I would like to make a few comments on the content of the agreements. I would also like to make a few suggestions for changes to the agreements.

2. Comment: Nature of Information Shared
Notwithstanding the title of the agreements, the fact that, under the terms of the current agreements, the NADF only shares knowledge, and not actual entries completely escaped me. Having realized this now, I feel that this limitation may prove to be a mixed blessing.

From a security and privacy standpoint, less is of course better. The privacy of entry information is only as great as the least trustworthy DSA that holds a complete copy of the entry, so the NADF practice effectively means that DMDs are free to implement any security measures necessary to preserve the privacy of information. From this standpoint therefore, the NADF restriction that the information shared is primarily knowledge is a Good Thing.

From the standpoint of Directory operation however, the single point of failure introduced by not sharing entry information is a major disadvantage of this aspect of the NADF agreements. In addition, experience with the Internet Directory pilot has shown very clearly that the sharing of entry information is central to obtaining adequate search performance from the system.

Although this shortcoming of the NADF agreement somewhat alleviated by the fact that DMDs may submit Knowledge Entries containing multiple logicalDSAReference values, I feel that it is still insufficient to guarantee system robustness and performance: entry information needs to be shared across DMDs against the day when a DMD will be unavailable.

3. Comment: Scope of Updates
I am extremely uncomfortable that the scope of the update information being exchanged between DMDs and the CAN is the entire DIT. While this is somewhat reasonable in the case of the CAN, I feel that this is completely unreasonable in the case of DMDs.

If a view of a DMD is taken as consisting of multiple DSAs, and not just one monolithic DSA containing the entire DIT, the processing of updates received from the CAN by such a DMD becomes extremely difficult. As any single DSA holds only a fragment of the DIT, and it cannot be assumed that the databases underlying the DSAs can be viewed as a logical whole outside of the X.500/DSA system, it becomes very difficult for any DSA to process updates on the entire DIT.

4. Comment: Form of Updates

The current structures currently defined for the exchange of information between the CAN and DMDs allows the specification of changes:

- as modifications to an entry, or by replacement of an entire entry
- on an individual entry or on an entire subtree

This seems a little more complex than it needs to be, in that it allows multiple ways both to reference an entry (as an individual entry, or as part of a subtree) and to update it (by modification or by replacement).

5. Comment: Sequence Numbering

I'm not completely convinced that the numbering scheme in the current NADF agreements is tenable.. It seems to me that the division between major and minor updates is somewhat artificial in that it really shouldn't make a difference to the revision state of the data whether the last operation performed on the data was the generation of a complete update or a delta update. In practice, the current sequencing scheme seems to introduce race conditions unnecessarily. For example:

- DMD A receives the delta update with sequence number n.m from the CAN and brings its local DIT up to date.
- After ascertaining from the CAN that it's local DIT is up to date, DMD A generates a delta update for the CAN. Since it's DIT revision level is n.m, the update it generates will have sequence number n.m, which it then sends to the CAN.
- In the meantime, DMD B requests a complete update from the CAN, resulting in the CAN's data now being at revision level (n+1).0
- The CAN rejects the update because not only is the sequence number wrong, but the DMD A appears to be a major revision behind.

Thus, through no fault of DMD A, it appears to be completely out of date! Also, it is to be noted that, the artificial sequence number notwithstanding, DMD A's DIT is in fact up to date. Generalizing to the n-case with multiple DMDs and multiple complete updates, it is possible to arrive at a scenario where some DMD An is some arbitrary number m of major revision levels behind the current state of the CAN.

The only possible solutions to this problem is to require the CAN to accept updates relative to any revision level of CAN state, past or present, or to ignore the problem, take the stance that the DMD lucked out, and needs to process the complete update (which it never wanted, note) before generating it's update. Neither option seems particularly attractive.

6. Suggestion: A New Model

Instead of the current symmetric model whereby DMDs send updates to the CAN, which simply turns about and propagates those same updates to other participating DMDs (but hopefully not to the submitting DMD!), I would like to suggest an asymmetric "push-pull" model where:

- DMDs continue to "push" updates to the CAN periodically. These updates would continue to contain all updates on the entire DIT.
- DMDs would "pull" updates from the CAN by sending requests containing
- a subtree specification defining the subtree for which updates are requested
- a time value such that any updates to the subtree specified received from any DMD since that the time specified should be returned.

This model involves a significant increase in the work undertaken by the CAN, which is no longer responsible just for performing syntax and consistency checks on updates it receives, but is now obliged to generate customized updates according to the specifications of the requests it receives.

However notice that the sequence numbering problem and the problem with DMDs that have multiple DSAs goes away. Since no use is made of artificial sequence numbers the problem of synchronization becomes much more manageable: although there are still synchronization problems due to latency in sending and receiving updates, the (currently larger) problem of synchronizing sequence numbers disappears.

In addition, since DMDs now specify the scope of updates, the problem of a mismatch between the scope of an update and the DIT fragment held by a DSA also goes away. It is the intent that DMDs submit update requests corresponding to the scheme by which the DIT is partitioned internally among the DSAs of the DMD.

So, it comes down to a tradeoff between increasing the burden on the CAN, something which the NADF has explicitly chosen to minimize; and dealing with the scope and synchronizations problems which I believe exist.

A suggested ASN.l definition for the DMD request protocol is provided in Appendix A to this document.

7. Suggestion: Form of Updates

While the current ASN.l definitions for knowledge and name sharing admit to a great deal of flexibility with respect to the way changes to the knowledge information in the DIT can be specified, this flexibility comes at the cost of an extremely complex specification. I would like to suggest that the NADF be a little less ambitious in terms of the flexibility desired in specifying DIT changes, and obtain in return a simpler specification.

At the most basic level, the objective of the NADF knowledge and name sharing agreements is to ensure that copies of the DIT held by DMDs are synchronized with respect to the listings and knowledge contained. Given this, I suggest that the form of updates consist simply of a list of entries that need to be changed, together with an indication of the operation required to change them. Specification of the entry to be changed is then accomplished by simply naming the entry in question.

I further suggest that entry changes be accomplished simply by replacing entries. While this may initially seem like an unnecessary increase in the size of updates due to the presence of extraneous attributes (extraneous only in the sense that they are unchanged), recall that the entries being updated are skeletal in nature, and contain a minimum of information besides attributes required to express knowledge. These entries would therefore tend to be small anyway, with the result that a major simplification is obtained for what will be a small increase in update size in the usual case.

A suggested ASN.1 specification for new constructs used in update can be found in Appendix B of this document.

8. Suggestion: Entry Sharing Agreements

For the operational performance reasons outlined in a previous section of this document, it is desirable for multiple copies of entry information to be available both within and without DMDs. This desire has to be balanced with the need to preserve the privacy of individual entries.

I suggest that a suitable balance point is for the NADF to leave the decision as to the extent to which entry sharing should occur to bilateral and/or multilateral agreements between DMDs, but encourage such sharing to the extent of providing the mechanism to allow such sharing.

Specifically, I suggest that procedures and the technical basis for entry sharing be developed collectively by the NADF for the use of DMDs that may choose to share entries outside of the confines of the DMD. As an aside, note that the sharing of entry information is essentially a process similar to the sharing of knowledge, and it would therefore advantageous to build on the work already done on knowledge and name sharing.

Appendix A
Form of DMD Requests

```
DMDRequest ::=
        SEQUENCE {
                baseTime     UTCTime,
                scope        SEQUENCE OF SubtreeSpec
        }
SubtreeSpec ::=
        SEQUENCE {
                subTreeRoot DistinguishedName,
                boundaries SEQUENCE OF RDNSequence
        }
```

Notes:

- The base time expresses the DIT revision level relative to which updates are to be expressed: any changes made to the DIT after that time stamp should be returned by the CAN to the DMD. So, a complete update is specified by using a very small base time value that precedes any possible change to the DIT (eg: 0).

- A subtree specification consists of a specification of the root of the subtree, together with nonleaf boundaries of the subtree. Leaves of the subtree are implicitly considered boundaries and need not be included in the subtree specification.

- The CAN returns the update requested using the constructs in Appendix B.

Appendix B
Form of Update Information

```
NADF-DSA-Info ::=
        SEQUENCE {
                baseTimeUTCTime,
                dmdNamePrintable String,
                dsaTableTable,
                informationInfo
        }
Info ::=
        SEQUENCE OF SEQUENCE {
                nameDistinguishedName, -- name of entry
                contentsSET OF Attribute - contents of entry
        }
```

Notes:

- The base time in the NADF-DSA-Info structure echoes the value in the DMD Request for CAN—> DMD Updates. For DMD –> CAN updates, it is the time at which the update was generated.
- The DSA Table has the same definition as in DF 277.
- The Info structure is now just a list of the entries that have changed. In the event that a subtree is being added, the entries that make up the subtree have to be occur in this list preorder. When entries are being deleted, they have to occur in the list postorder.
- The operation that needs to be performed to update the entry is implicit: if an entry that occurs in the Info list does not occur in the DIT, and the contents of the entry in the Info list is nonnull, the operation is 'add'; if the entry in the Info list

does occur in the DIT, and the contents of the entry in the Info list is nonnull, the contents of the entry in the DIT is replaced by the contents of the entry in the Info list; and if the entry in the Info list has no attributes (contents of the entry are null), the entry is deleted. The modifyDN operation needs to be performed by a delete followed by an add.

☆U.S. GOVERNMENT PRINTING OFFICE: 1995-610-126-50111

DISTRIBUTION LIST

addresses	number of copies
ATTN: SENATRO J. IUORNO RL/C3BC 525 BROOKS ROAD GRIFFISS AFB NY 13441-4505	5
NYSERNET, INC. SUITE 103 200 ELWOOD DAVIS ROAD LIVERPOOL, NY 13088-6147	5
RL/SUL TECHNICAL LIBRARY 26 ELECTRONIC PKY GRIFFISS AFB NY 13441-4514	1
ADMINISTRATOR DEFENSE TECHNICAL INFO CENTER DTIC-FDAC CAMERON STATION BUILDING 5 ALEXANDRIA VA 22304-6145	2
ADVANCED RESEARCH PROJECTS AGENCY 3701 NORTH FAIRFAX DRIVE ARLINGTON VA 22203-1714	1
COMMANDING OFFICER NCCOSC RDTE DIVISION CODE 0274B, TECH LIBRARY 53560 HULL STREET SAN DIEGO CA 92152-5001	1
AEDC LIBRARY TECH FILES/MS-100 ARNOLD AFB TN 37389	1
COMMANDER/USAISC ATTN: ASOP-DO-TL BLDG 61801 FT HUACHUCA AZ 85613-5000	1

ESC/AV 1
20 SCHILLING CIRCLE
HANSCOM AFB MA 01731-2816

FL 2807/RESEARCH LIBRARY 1
OL AA/SULL
HANSCOM AFB MA 01731-5000

TECHNICAL REPORTS CENTER 1
MAIL DROP D130
BURLINGTON ROAD
BEDFORD MA 01731

DEFENSE TECHNOLOGY SEC ADMIN (DTSA) 1
ATTN: STTD/PATRICK SULLIVAN
400 ARMY NAVY DRIVE
SUITE 300
ARLINGTON VA 22202

ARPA/ASTO 3
ATTN: DR. STUART MILNER
3701 N. FAIRFAX DRIVE
ARLINGTON, VA 22203-1714

ARPA/CSTO 3
ATTN: MAJ MICHAEL ST JOHNS
3701 N. FAIRFAX DRIVE
ARLINGTON, VA 22203-1714

MISSION
OF
ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

a. Conducts vigorous research, development and test programs in all applicable technologies;

b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

c. Provides a full range of technical support to Air Force Materiel Command product centers and other Air Force organizations;

d. Promotes transfer of technology to the private sector;

e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.